# UC San Diego

## UC San Diego Electronic Theses and Dissertations

### Title

Large-Scale Trust-Region Methods and Their Application to Primal-Dual Interior-Point Methods

### Permalink

https://escholarship.org/uc/item/8s29n42d

### Author

Guldemond, Alexander

### Publication Date

2023

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA SAN DIEGO

# Large-Scale Trust-Region Methods and Their Application to Primal-Dual Interior-Point Methods

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Mathematics

by

Alexander Guldemond

Committee in charge:

Professor Philip Gill, Chair
Professor Michael Holst
Professor John Hwang
Professor Rayan Saab

2023

Copyright

Alexander Guldemond, 2023

All rights reserved.

The Dissertation of Alexander Guldemond is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

## VITA

2016 Bachelor of Arts, Boston University, Boston

2020 Master of Arts, University of California San Diego

2023 Doctor of Philosophy, University of California San Diego

## FIELDS OF STUDY

Major Field: Mathematics (Specialization or Focused Studies)

Studies in Applied Mathematics
Professors Philip Gill

ABSTRACT OF THE DISSERTATION

Large-Scale Trust-Region Methods and Their Application to Primal-Dual Interior-Point Methods

by

Alexander Guldemond

Doctor of Philosophy in Mathematics

University of California San Diego, 2023

Professor Philip Gill, Chair

Trust-region methods are amongst the most commonly used methods in unconstrained mathematical optimization. Their impressive performance and sound theoretical guarantees make them suitable for a wide range of problem types. However, the computational complexity of existing methods for solving the trust-region subproblem prevents trust-region methods from being widely used in large-scale problems in both unconstrained and constrained settings. This dissertation introduces and analyzes three novel methods for solving the trust-region subproblem for large-scale constrained optimization problems. Convergence rates and proofs are presented where applicable. Furthermore, a trust-region approach is developed for the recently introduced all-shifted primal-dual penalty-barrier method for solving nonconvex, constrained optimization problems.

The three trust-region algorithms introduced are the shifted and inverted generalized Lanczos trust region algorithm, the locally optimal preconditioned conjugate gradient trust region, and the Jacobi-Davidson QZ trust region algorithm. Each new method exhibits improved performance over the existing

standard methods and is best suited for problems too large for the traditional methods to handle efficiently. Furthermore, each method exhibits particular benefits for differently scaled problems.

# Introduction

## 0.1 Overview

Mathematical optimization, sometimes called mathematical programming, is the selection of the best element from a set of available alternatives. Although there are many different fields within optimization, the subject can generally be divided into discrete and continuous optimization. Optimization problems arise in every quantitative discipline, including but not limited to computer science, engineering, operations research, economics, and data science. The study of optimization relies on formulating a mathematical model of a given problem. Optimizing the model then means finding a point that either minimizes or maximizes the model. These two formulations are interchangeable: maximizing a function is equivalent to minimizing the negative of the function. Discrete optimization refers to problems in which the set of allowable points is discrete, while continuous optimization refers to problems in which the set of permissible points is continuous.

Continuous optimization can further be broken down into unconstrained and constrained optimization classes. In unconstrained optimization, all points in the domain of the function $f(x)$ to be minimized, the *objective function*, are possible solutions. In constrained optimization, the set of allowable points, called the *feasible set*, excludes certain points. The mathematical model of the problem represents the feasible set by a set of equality and inequality conditions.

The field of continuous optimization can further be broken down into the classes of convex and nonconvex optimization. Convex optimization is only concerned with convex objective functions and constraint sets. Nonconvex optimization, on the other hand, does not make any assumptions of convexity. As a result, nonconvex algorithms are generally designed to perform well when applied to convex problems, assuming that the convex problem in question satisfies whatever properties are required of the given algorithm. On the other hand, algorithms for convex problems cannot be applied to nonconvex problems.

Every method discussed in this thesis is designed to solve continuous nonconvex optimization problems and is both iterative and approximate in nature. If $x^\star$ is the unknown solution, then a

sequence $\{x_k\}_{k=1}^{\infty}$ is generated such that each subsequent point in the sequence is a better estimate of $x^\star$. Theoretically, these sequences are infinite. In practice, all algorithms terminate once the approximate solution $x_k$ satisfies some optimality conditions to sufficient accuracy. These optimality conditions differ depending on the type of problem being solved and play an essential role in any implementation of an algorithm and its accompanying theoretical discussion. Such a sequence is generated by formulating a local model of the objective function at the current estimate $x_k$, and computing an update of the form $x_{k+1} = x_k + p_k$, where $p_k$ is a vector which decreases the objective value and is inferred from the local model.

One of the most popular implementations of this general procedure is the trust-region method. In the trust-region method, the local model function is typically taken to be the second-order Taylor series of the objective about the current estimate $x_k$. The model is then minimized subject to the constraint that the minimum lie in a convex subset of the feasible set called the trust region. This simple procedure has enormous practical success and has strong theoretical guarantees. However, the application of trust-region algorithms to large-scale problems is made difficult by the fact that the minimization of the quadratic model subject to the trust-region constraint, called the *trust-region subproblem*, is itself a constrained potentially nonconvex optimization problem. Although some methods exist for solving large-scale trust-region subproblems, they perform poorly for particular problems, especially in constrained cases. This thesis introduces three novel approaches to solving the large-scale trust-region subproblem. These methods are designed to apply to the constrained case. None of them is claimed to be the best algorithm in every case. However, between these three methods and the existing ones reviewed here, all but the most extreme problems should be amenable to the trust-region method.

Applying the trust-region strategy to constrained problems presents several difficulties. One of the most widely used classes of algorithms for solving constrained problems is the class of interior-point methods. As the name implies, interior-point methods generate a sequence of points $\{x_k\}$ that lies strictly within the interior of the feasible set. This sequence is formed by adding a term to the objective function that tends towards infinity as the sequence approaches the boundary. As the method continues, this barrier term is modified to allow points to get closer to the boundary. As the solution to constrained problems typically lies on the boundary of the feasible set, the sequence generated by interior-point methods will only ever become arbitrarily close to the true solution. In [13], a new class of interior-point methods called shifted barrier methods are introduced, which shifts the boundary of the feasible set, allowing the sequence to fall on the feasible set's boundary. This new approach has been shown to improve traditional interior-point methods significantly. This dissertation presents a shifted barrier method with a trust-region

approach using the new algorithms for solving the trust-region subproblem.

This dissertation is organized into six chapters. Chapter 1 begins with a review of the necessary results from linear algebra used throughout this thesis, including some less commonly known procedures, such as the relationship between the Lanczos process and the preconditioned conjugate gradient algorithm. Chapter 2 reviews results and techniques from unconstrained optimization, with particular emphasis on the trust-region method. Chapter 3 examines results and techniques from constrained optimization, with particular emphasis on interior-point methods. Chapter 4 reviews existing algorithms for solving the trust-region subproblem and introduces the three new algorithms presented here. Chapter 5 discusses the shifted barrier trust-region method. Finally, chapter 6 presents numerical results.

## 0.2 Contributions of This Dissertation

Chapter 2 introduces a more general set of assumptions under which the trust-region method is guaranteed to converge. Typically speaking, the convergence results of trust-region algorithms assume that the method used to compute the minimizer of the trust-region subproblem computes a point that is as good as the negative gradient. This assumption is difficult to directly verify for the novel methods presented in this dissertation. The more general result is that the method used to compute the minimizer of the trust-region subproblem computes a point that is as good as any arbitrary steepest-descent direction. Each of the techniques presented begins by first taking a different steepest-descent direction as the initial estimate, making this assumption trivial to verify.

Chapter 4 introduces the three novel algorithms for solving the trust-region subproblem. The first of these methods utilizes a shifted and inverted Lanczos process to transform a high-dimensional, potentially ill-conditioned trust-region subproblem into a well-conditioned, low-dimensional problem that can be trivially solved with existing methods. A technique for warm-starting this method, given an initial approximation, is presented as well. As trust-region methods solve a sequence of trust-region subproblems, the ability to utilize prior information from a previous subproblem is crucial to maintain rapid performance. Additionally, the unstable nature of the Lanczos process necessitates the ability to restart the process every so often. This enables the new algorithm to rapidly find the solution with previous information to a high degree of accuracy. The second method presented is an application of a conjugate-gradient style algorithm to the trust-region subproblem with a locally-optimal update taken at each iteration. Equivalence between the trust-region subproblem and an unconstrained problem is established. This equivalence allows results from unconstrained nonlinear conjugate-gradient algorithms to be applied to the method, guaranteeing convergence. This second method differs from the first in that a fixed dimensional

subproblem is solved at each iteration to generate the locally optimal improvement to the approximate solution. Additionally, unlike the first method, this method can use any preconditioning technique, or potentially no preconditioning, to correct for any ill-conditioning. The final method discussed is based on the Jacobi-Davidson QZ algorithm for solving generalized eigenvalue problems and is specifically designed for a class of trust-region problems arising in constrained optimization called doubly-augmented trust-region problems. The relationship between doubly-augmented matrices and regularized saddle-point matrices is exploited to create a generalized algorithm that can use preconditioners that are not required to be positive definite. For doubly-augmented problems, this implies that no explicit matrix factorizations are necessary. Experiments reveal that for most practical problems, the first two methods perform considerably better than existing methods. The last method, on the other hand, has demonstrated rapid convergence on artificially generated doubly-augmented trust-region problems that are too large and ill-conditioned to utilize any other method.

Chapter 5 presents a primal-dual penalty-barrier trust-region method for solving general nonlinear constrained optimization problems utilizing the shifting strategy first introduced in [13] and [17]. Large-scale trust-region methods typically suffer from the need to solve the trust-region subproblem at each iteration, and thus techniques such as line searches are often preferred. However, the new methods introduced in Chapter 4 make a large-scale trust-region algorithm far more feasible. This is particularly useful in interior-point methods, where the barrier terms cause the trust-region subproblems to be increasingly ill-conditioned as the solution is approached. The shifting of the primal-dual feasible set helps reduce ill-conditioning's influence, but it does not remove it entirely. The convergence of the trust-region method for minimizing the penalty-barrier function is established. Additionally, global convergence is established following the results in [13].

## 0.3 Notation

Given a set of vector $\{x_1, x_2, \ldots, x_m\}$, where $x_i \in \mathbb{R}^{n_i}$ for some $n_i \in \mathbb{N}$ for all $i = 1, \ldots, m$, the vector $x$ consisting of the concatenation of all vectors $x_1, \ldots, x_m$ is given by $(x_1, \ldots, x_m)$. The subscript $k$ appended to a vector will typically denote its value during the $k$-th iterate of an algorithm or the $k$-th value in a sequence. Square brackets and subscripts shall denote a particular entry in a vector, i.e., $[x_k]_i$ shall denote the $i$-th entry of the vector $x_k$. If $x$ is not a member of a sequence of vectors, then the square brackets are dropped. Given a matrix $A \in \mathbb{R}^{n \times m}$, pairs of subscripts shall denote a particular entry of the matrix $A$, i.e., $A_{i,j}$ denotes the entry of $A$ in the $i$-th row and the $j$-th column. Given two vectors $x$ and $z$ of the same dimension, the vector with $i$-th component $[x]_i[z]_i$ is denoted by $x \cdot y$. Similarly,

$\min\{x, z\}$ and $\max\{x, z\}$ shall denote the vector whose $i$-th component is $\min\{[x]_i, [z]_i\}$ or $\max\{[x]_i, [z]_i\}$, respectively. The vectors $e_i$ shall denote the vector of all zeros except for the $i$-th entry, which has a 1, and the vector $e$ shall denote the vector of all 1s. The matrix $I$ shall denote the identity matrix. The context makes the size of $e_i$, $e$, and $I$ apparent. If the size of $I$ is not apparent from the context, a subscript is appended to denote its shape, i.e., $I_n$ denotes the $n \times n$ identity matrix. The symbol 0 denotes the scalar 0, the vector of all zeros, and the matrix of all zeros, where the size and shape are apparent from the context. Arbitrary matrix and vector norms are denoted with $||\cdot||$. The typical $p$-norms, and their corresponding induced matrix norms, are denoted with $||\cdot||_p$. The notation $\{||\cdot||_k\}_{k\in\mathcal{K}}$ denotes a sequence of norms indexed by some set $\mathcal{K}$. The inertia of a symmetric matrix $A$, denoted by $\text{In}(A)$, is the integer triple $(n_+, n_-, n_0)$ of positive, negative, and zero eigenvalues of $A$. The ordered pair $(A, B)$ denotes the matrix pencil defined by matrices $A$ and $B$. Given a function $f(x) : \mathbb{R}^n \to \mathbb{R}$, the gradient of $f$ at a point $x$ is denoted as $\nabla f(x)$, and the Hessian matrix is denoted as $\nabla^2 f(x)$. The symbol $g(x)$ is often used to denote the gradient $\nabla f(x)$, and, given a sequence of points $\{x_k\}_{k\in\mathcal{K}}$, $\{g_k\}_{k\in\mathcal{K}}$ denotes the sequence $\{\nabla f(x_k)\}_{k\in\mathcal{K}}$. The matrix $J(x)$ denotes the Jacobian of a vector-valued function $c(x)$. The Lagrangian of a constrained optimization problem is $L(x, y)$. The Hessian of the Lagrangian with respect to the primal variables is denoted as $H(x, y)$ and is given by $\nabla^2_{x,x} L(x, y)$. Let $\{x_k\}_{k\in\mathcal{K}}$ be a sequence of scalars, vectors, or matrices, and $\{b_k\}_{k\in\mathcal{K}}$ a sequence of positive scalars. The notation $x_k = \mathcal{O}(b_k)$ implies that there exists a positive scalar $\gamma$ such that $||x_j|| \leq \gamma b_j$ for all $j \in \mathcal{K}$. If there exists a sequence $\{\gamma_k\}$ converging to zero such that $||x_j|| \leq \gamma_j b_j$, then $x_j = o(b_j)$. If there exists a sequence $\{\sigma_k\}$ converging to zero and a positive constant $b$ such that $b_j > b\sigma_j$, then $b_j = \Omega(\sigma_j)$. The symbols $\succ$, $\succeq$, $\prec$, and $\preceq$ denote the Loener ordering on square, symmetric matrices, i.e., $A \succ B$ if and only if $A - B$ is positive definite. The statement $A \succ 0$ implies that $A$ is positive definite, and $A \succeq 0$ implies that $A$ is positive semi-definite. Given a norm $||\cdot||$, the dual is denoted by $||\cdot||_*$. Given two vector norms $||\cdot||_\alpha$ and $||\cdot||_\beta$, the matrix norm induced by these two norms is denoted as $||\cdot||_{\alpha,\beta}$, i.e., $||A||_{\alpha,\beta} = \max_{x\neq 0} ||Ax||_\beta / ||x||_\alpha$. Given an unadorned vector norm $||\cdot||$, the matrix norm induced by $||\cdot||$ and $||\cdot||_\star$ is denoted as $||A||$. If $B$ is a symmetric positive definite matrix, then $\langle\cdot,\cdot\rangle_B$ and $||\cdot||_B$ denote the inner-product and norm induced by $B$, respectively, i.e. $\langle x, y\rangle_B = y^{\mathrm{T}} B x$, and $||x||_B = \sqrt{x^{\mathrm{T}} B x}$. The symbols $>$, $\geq$, $<$, and $\leq$ applied to vectors $x$ and $y$ are applied element-wise, i.e. $x < y$ is true if $[x]_i < [y]_i$ for all indices $i$.

# Chapter 1

# Linear Algebra

This section discusses some fundamental results of Linear Algebra that are used throughout this dissertation. Most of these results shall be presented without proof, as the proofs can be found in any graduate text on linear algebra.

## 1.1 Symmetric Positive Definite Matrices

**Definition 1.1.1** (Symmetric Positive Semidefinite Matrices)**.** Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix, i.e., $A = A^{\mathrm{T}}$. $A$ is called positive semidefinite if

$$x^{\mathrm{T}}Ax \geq 0$$

for all $x \in \mathbb{R}^n$.

**Definition 1.1.2** (Symmetric Positive Definite Matrices)**.** A symmetric positive semidefinite matrix $A \in \mathbb{R}^{n\times n}$ is called symmetric positive definite if $x^{\mathrm{T}}Ax > 0$ for all $x \in \mathbb{R}^n$ not equal to zero.

**Lemma 1.1.1** (Spectral Characterization of Positive Definite Matrices)**.** *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive semidefinite. Let $\lambda$ be an eigenvalue of $A$, i.e., there exists a vector $v \neq 0$ such that $Av = \lambda v$. Then $\lambda$ is real and nonnegative. If $A$ is a symmetric positive definite matrix, then $\lambda$ is strictly positive. Conversely, a symmetric matrix $A$ is positive semidefinite (definite) if all eigenvalues of $A$ are real and nonnegative (positive).*

**Theorem 1.1.2** ([21], Theorem 7.2.6)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite matrix. Let $k \in \{2, 3, \ldots\}$.*

1. *There is a unique symmetric positive semidefinite matrix $B$ such that $B^k = A$.*

2. *There is a polynomial $p$ with real coefficients such that $B = p(A)$. Consequently, $B$ commutes with any matrix that commutes with $A$.*

3. $\text{range}(A) = \text{range}(B)$, *so* $\text{rank}(A) = \text{rank}(B)$.

**Theorem 1.1.3** (Cholesky decomposition, [21], Theorem 7.2.7)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. Then $A$ is positive semidefinite (respectively, positive definite) if and only if there is an upper triangular matrix $R \in \mathbb{R}^{n\times n}$ with nonnegative (respectively, positive) diagonals entries such that $A = R^{\mathrm{T}}R$. If $A$ is positive definite, then $R$ is unique.*

## 1.2 Congruence

**Definition 1.2.1** (Congruence Relation)**.** Let $A$, $B \in \mathbb{R}^{n\times n}$ be given. If there exists a nonsingular matrix $S$ such that $B = SAS^{\mathrm{T}}$, then $B$ is said to be *congruent* to $A$.

**Theorem 1.2.1** ([21], Theorem 4.5.5)**.** *Congruence is an equivalence relation.*

The objective for examining this equivalence relation is to be able to infer properties of the eigenvalues of a matrix $A$, for which the eigenvalues themselves may be infeasibly difficult to calculate, by making observations of the spectrum of a congruent matrix $B$. For this, the *inertia* of a symmetric matrix is defined to be

**Definition 1.2.2** (Inertia)**.** Let $A \in \mathbb{R}^{n\times n}$ be symmetric. The *inertia* of $A$ is the ordered triple

$$\text{In}(A) = (m_+, m_-, m_0),$$

where $m_+$ denotes the number of positive eigenvalues of $A$, $m_-$ the number of negative eigenvalues, and $m_0$ the number of eigenvalues identically zero,

The following result will be used extensively in the discussion of constrained optimization.

**Theorem 1.2.2** (Sylvester's Law of Inertia, [21], Theorem 4.5.8)**.** *Symmetric matrices $A$, $B \in \mathbb{R}^{n\times n}$ are congruent if and only if they have the same inertia, that is, if and only if they have the same number of positive and negative eigenvalues.*

Sylvester's Law of Inertia states that a congruence transformation can be applied to a matrix $A$ to yield a matrix $B$ whose inertia is readily apparent in order to learn the inertia of $A$. One particular congruence transformation is that which yields the Schur complement.

**Definition 1.2.3** (Schur Complement)**.** Let a symmetric matrix $M \in \mathbb{R}^{n\times n}$ be partitioned as

$$M = \begin{pmatrix} A & B \\ B^{\mathrm{T}} & C \end{pmatrix}.$$

Assume that $A$ is nonsingular. Let

$$L = \begin{pmatrix} I & 0 \\ -B^{\mathrm{T}}A^{-1} & I \end{pmatrix}.$$

Then $\mathrm{In}(M) = \mathrm{In}(N)$, where

$$N = L^{\mathrm{T}}ML = \begin{pmatrix} A & 0 \\ 0 & C - B^{\mathrm{T}}A^{-1}B \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & S \end{pmatrix},$$

where $S = C - B^{\mathrm{T}}A^{-1}B$ is referred to as the *Schur Complement*.

A second strategy that is ubiquitous in constrained optimization is the *symmetric indefinite factorization*, sometimes referred to as the *inertia revealing factorization*.

**Theorem 1.2.3** (Symmetric Indefinite Factorization)**.** *Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and symmetric. Then there exists a permutation matrix $P$, a unit lower triangular matrix $L$, and a block diagonal matrix $D$ with $1\times 1$ and $2\times 2$ blocks such that*

$$PAP^{\mathrm{T}} = LDL^{\mathrm{T}},$$

*with every $2\times 2$ block of $D$ having one positive and one negative diagonal.*

There are many different strategies for forming the symmetric indefinite factorization of a matrix. More specifically, there are many different strategies for forming the permutation matrix $P$. In small to medium-scale problems, $P$ is chosen to preserve the stability of the factorization. In large, sparse cases, there exists the additional goal of choosing $P$ such that the factors $L$ maintain the sparsity of $A$ without sacrificing stability. There exists a large array of available implementations of $LDL^{\mathrm{T}}$ factorizations to choose from. Bunch and Kaufman describe a pivoting strategy in [6]. The symmetric indefinite factorization, when applied to a positive definite matrix, is sometimes referred to as the *square root free Cholesky decomposition*, as the implementation requires no square roots. In this case, it is guaranteed that $D$ is positive definite and diagonal. The permutation matrix, in this case, can always be taken to be the identity matrix, however, other choices are used to improve stability.

## 1.3 Vector Norms and Matrix Norms

**Theorem 1.3.1** (Norm Equivalence on Finite Dimensional Vector Spaces, [21], Corollary 5.4.6)**.** *Let $V$ be a finite dimensional vector space, and let $||\cdot||_\alpha$ and $||\cdot||_\beta$ be two norms on $V$. The norms $||\cdot||_\alpha$ and $||\cdot||_\beta$ are equivalent, i.e., there exists constants $c_1 > 0$, $c_2 > 0$ such that*

$$c_1||x||_\alpha \leq ||x||_\beta \leq c_2||x||_\alpha$$

*for all $x \in V$.*

Note that Theorem 1.3.1 also implies that

$$\frac{1}{c_2}||x||_\beta \leq ||x||_\alpha \leq \frac{1}{c_1}||x||_\beta$$

for all $x \in V$.

**Definition 1.3.1** (Dual Norm)**.** Let $V$ be a vector space equipped with norm $||\cdot||$. Let $v^*$ be the dual space of $V$, i.e., the space of all linear functionals on $V$. The *dual norm* of $||\cdot||$, denoted as $||\cdot||_\star$, is given by

$$||y||_\star = \max_{x \neq 0} \frac{|y(x)|}{||x||}$$

for all $y \in v^*$.

One immediate observation that can be made is that for any $x \in V$, $y \in v^*$, $|y(x)| \leq ||x||\ ||y||_\star$. If $V$ is an inner product space equipped with inner product $\langle \cdot, \cdot \rangle$, then $|\langle x, y \rangle| \leq ||x||\ ||y||_\star$. If $V = \mathbb{R}^n$, then this becomes $|y^{\mathrm{T}}x| \leq ||x||\ ||y||_\star$.

For $1 \leq p \leq \infty$, let $||x||_p$ denote the $p$-norm $||x||_p = (\sum_{i=1}^n |x_i^p|)^{1/p}$, with $||x||_\infty = \max_{i \in \{1,\ldots,n\}} |x_i|$. Let $B \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix. $B$ then induces an inner product and a norm denoted by, respectively

$$\langle x, y \rangle_B = y^{\mathrm{T}}Bx, \quad \text{and} \quad ||x||_B = \sqrt{x^{\mathrm{T}}Bx}$$

for all $x, y \in \mathbb{R}^n$.

**Theorem 1.3.2.** *Let $B \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix. The dual norm of $||\cdot||_B$, denoted as $||\cdot||_{B\star}$, is given by $||\cdot||_{B^{-1}}$.*

*Proof.* Due to the fact that $B$ is a symmetric positive definite matrix, $B$ is nonsingular. Then

$$||y||_{B\star} = \max_{x\neq 0} \frac{|y^{\mathrm{T}}x|}{\sqrt{x^{\mathrm{T}}Bx}}.$$

Let $B^{1/2}$ denote the unique, positive definite square root of $B$. Then

$$\begin{aligned} ||y||_{B\star} &= \max_{x\neq 0} \frac{|y^{\mathrm{T}}x|}{\sqrt{x^{\mathrm{T}}Bx}} \\ &= \max_{x\neq 0} \frac{|y^{\mathrm{T}}x|}{\sqrt{x^{\mathrm{T}}B^{1/2}B^{1/2}x}} \\ &= \max_{v\neq 0} \frac{|y^{\mathrm{T}}(B^{1/2})^{-1}v|}{\sqrt{v^{\mathrm{T}}v}}. \end{aligned}$$

By the Cauchy-Schwartz inequality $|y^{\mathrm{T}}(B^{1/2})^{-1}v| \leq ||(B^{1/2})^{-1}y||_2||v||_2 = ||y||_{B^{-1}}||v||_2$ for all $y, v \in \mathbb{R}^n$. This upper bound is achieved with $v = (B^{1/2})^{-1}y$. Therefore, $||y||_{B\star} = ||y||_{B^{-1}}$. □

**Corollary 1.3.2.1.** *Let $B \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. For all $x, y \in \mathbb{R}^n$,*

$$|y^{\mathrm{T}}x| \leq ||x||_B||y||_{B^{-1}}.$$

Let $||\cdot||$ be an arbitrary norm on $\mathbb{R}^n$. This norm $||\cdot||$ can be used to construct a corresponding matrix norm on the vector space of square matrices $\mathbb{R}^{n\times n}$.

**Definition 1.3.2** (Induced Norms)**.** Let $V$ be a vector space equipped with a norm $||\cdot||$. Let $A : V \to V$ be a linear operator. Then the induced norm of $A$ is given by

$$||A|| = \max_{x\neq 0} \frac{||Ax||}{||x||}.$$

This definition can be extended to the case where $A : V \to W$, where $V$ is a vector space equipped with norm $||\cdot||_\alpha$, and $W$ is a vector space equipped with norm $||\cdot||_\beta$.

**Definition 1.3.3.** Let $(V, ||\cdot||_\alpha)$ and $(W, ||\cdot||_\beta)$ be two normed linear spaces, and let $A$ be a linear operator from $V$ to $W$. Then the norm of $A$ induced by the norms $||\cdot||_\alpha$ and $||\cdot||_\beta$ is given by

$$||A||_{\alpha,\beta} = \max_{x\neq 0} \frac{||Ax||_\beta}{||x||_\alpha}.$$

This section is concluded with a common result about the induced 2-norm on symmetric matrices.

**Lemma 1.3.3.** *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. Let $\lambda_{\max}$ denote the eigenvalue of $A$ with the largest magnitude. Then*

$$||A||_2 = \max_{x\neq 0} \frac{||Ax||_2}{||x||_2} = \max_{x\neq 0} \frac{x^{\mathrm{T}}Ax}{x^{\mathrm{T}}x} = \lambda_{\max}.$$

## 1.4 Condition Numbers

When analyzing the stability and performance of various numerical methods for solving linear algebra problems, one frequently encounters the notion of the *condition number* of a matrix $A$.

**Definition 1.4.1** (Condition Number)**.** Given a matrix norm $||\cdot||$, the *condition number* of a nonsingular matrix $A$ is given by

$$\kappa(A) = ||A||\ ||A^{-1}||.$$

A matrix with a large condition number is said to be *ill-conditioned*. Conversely, if the condition number is close to 1, then the matrix is considered to be *well-conditioned*. Clearly, the condition number of a matrix depends on the norm. Typically, when referring to the condition number, the 2-norm is meant implicitly. The performance of some algorithms can be improved by choosing a norm ahead of time which reduces the condition number of a particular matrix.

Consider an orthogonal matrix $Q$. As $||Qx||_2 = ||x||_2$ for all $x \in \mathbb{R}^n$, and $Q^{-1} = Q^{\mathrm{T}}$, it is clear that

$$\kappa(Q) = ||Q||_2\ ||Q^{\mathrm{T}}||_2 = 1.$$

Therefore, a unitary matrix $Q$ is perfectly conditioned in the 2-norm. Conversely, consider the matrix $H_n$ given by

$$(H_n)_{i,j} = \frac{1}{i+j-1}, \quad 1 \leq i,j \leq n.$$

This matrix is referred to as the *Hilbert matrix* of order $n$ and is an extreme example of an ill-conditioned matrix. For instance, even at the low dimension of $n = 12$, it can be shown that $\kappa(H_{12}) \approx 1.6 \times 10^{16}$, and that the standard method of Gaussian elimination applied to the equation $H_{12}x = b$ fails in double-precision arithmetic.

If the matrix norm in question is induced by two norms instead of just one, i.e.,

$$||A||_{\alpha,\beta} = \max_{x\neq 0} ||Ax||_\beta / ||x||_\alpha,$$

the condition number is more appropriately defined as

$$\kappa(A) = ||A||_{\alpha,\beta}||A^{-1}||_{\beta,\alpha},$$

reflecting the fact that in this case, $A$ is an operator from one normed space to a different normed space. To see this, consider the nonsingular matrix $A$ as an operator from $(\mathbb{R}^n, ||\cdot||_\alpha)$ to $(\mathbb{R}^n, ||\cdot||_\beta)$. Consider the system $Ax = b$, and the perturbed system $A(x + \delta x) = b + \delta b$. Then

$$||Ax||_\beta = ||b||_\beta, \quad \text{and} \quad ||\delta x||_\alpha = ||A^{-1}\delta b||_\alpha.$$

From the first of these two equations, it holds that

$$||b||_\beta \leq ||A||_{\alpha,\beta}||x||_\alpha,$$

and from the second,

$$||\delta x||_\alpha \leq ||A^{-1}||_{\beta,\alpha}||\delta b||_\beta.$$

Combining these two results gives

$$\frac{||\delta x||_\alpha}{||x||_\alpha} \leq \kappa(A)\frac{||\delta b||_\beta}{||b||_\beta},$$

where $\kappa(A) = ||A||_{\alpha,\beta}||A^{-1}||_{\beta,\alpha}$.

## 1.5 Generalized Eigenvalues and Eigenvectors

**Definition 1.5.1.** Let $A, B \in \mathbb{R}^{n\times n}$ be two square matrices. The generalized eigenvalues and eigenvectors of matrices $A$ and $B$ are the solution pairs $(\lambda, v)$ of the *generalized eigenvalue* problem

$$Av = \lambda Bv,$$

where $v \in \mathbb{C}^n$ and $\lambda \in \mathbb{C} \cup \infty$. The solutions $\lambda$ and $v$ are also referred to as an eigenvalue and an eigenvector, respectively, of the *matrix pencil* $(A, B)$.

The term *matrix pencil* may also be used to describe the family of matrices of the form $A - \lambda B$. The case $\lambda = \infty$ corresponds to the case where $B$ is singular, and $v \in \text{null}(B)$. Some authors prefer to

avoid the inclusion of $\lambda = \infty$, and so define the generalized eigenvalue problem as finding solutions to

$$\alpha A v = \beta B v,$$

where $\alpha$, $\beta \in \mathbb{C}$ are normalized so that $|\alpha|^2 + |\beta|^2 = 1$. In this dissertation, only nonsingular choices of $B$ are considered, and therefore the first formulation of the generalized eigenvalue problem is preferred.

A matrix pencil $(A, B)$ is called *regular* if $\det(A - \lambda B)$ is not identically equal to zero. In this case, the following result holds.

**Theorem 1.5.1** (Generalized Schur Decomposition, [21], Theorem 2.6.1)**.** *Let $A$, $B \in \mathbb{R}^{n\times n}$. Then there exist unitary matrices $Q, Z \in \mathbb{C}^{n\times n}$ such that $A = QSZ^\star$ and $B = QTZ^\star$. The eigenvalues of $(A, B)$ are the ratio of the diagonals if the matrices $S$ and $T$, i.e., $\lambda_i = S_{i,i}/T_{i,i}$ for each index $i$.*

This decomposition is often referred to as the QZ decomposition. There are instances in which a problem only calls for the real eigenvalues of a matrix pencil. In this case, it is more convenient to work with the real version of the generalized Schur decomposition.

**Theorem 1.5.2** ([21], Theorem 2.6.2)**.** *Let $A, B \in \mathbb{R}^{n\times n}$. There exist real orthogonal matrices $Q, Z \in \mathbb{R}^{n\times n}$ such that $A = QSZ^{\mathrm{T}}$ and $B = QTZ^{\mathrm{T}}$, where $T$ is a real upper triangular matrix with nonnegative diagonals, and $S$ is real and upper quasitriangular, i.e., block upper triangular with blocks of size $1 \times 1$ and $2 \times 2$, in which the $2 \times 2$ blocks correspond to complex eigenvalues.*

One fairly intuitive method for dealing with generalized eigenvalue problems is to transform them into a standard eigenvalue problem. Consider the generalized eigenvalue problem $Av = \lambda Bv$ for matrices $A, B \in \mathbb{R}^{n\times n}$. Assume the pencil $(A, B)$ is regular so that there exists a shift $\mu$ such that $A + \mu B$ is nonsingular. Then the generalized eigenvalue problem is equivalent to the problem $(A+\mu B)v = (\lambda+\mu)Bv$, which in turn is equivalent to,

$$(A + \mu B)^{-1} B v = \frac{1}{\lambda + \mu} v.$$

Many algorithms for finding generalized eigenvalues of $(A, B)$ begin by performing this transformation for some $\mu$ with $-\mu$ close to, but not identical to, the desired eigenvalue.

If $B$ is a symmetric positive definite matrix, there is a simpler transformation that can be applied. Let $R^{\mathrm{T}}R = B$ be the Cholesky decomposition of $B$. Then

$$Ax = \lambda Bx = \lambda R^{\mathrm{T}} R x.$$

If $y = Rx$, then

$$AR^{-1}y = \lambda R^{\mathrm{T}}y, \quad \text{or equivalently} \quad R^{-\mathrm{T}}AR^{-1}y = \lambda y.$$

If $R^{-\mathrm{T}}AR^{-1}$ is normal, then the standard spectral theorem applies, and $\lambda$ is guaranteed to be real. In fact, if $A$ is symmetric, the following generalization of the spectral decomposition exists.

**Theorem 1.5.3.** *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and $B \in \mathbb{R}^{n\times n}$ be symmetric positive definite. Then there exists a $B$-orthogonal matrix $U$, i.e., $U^{\mathrm{T}}BU = I$, a $B^{-1}$-orthogonal matrix $V$, and a real diagonal matrix $\Lambda$ such that*

$$U^{\mathrm{T}}AU = \Lambda, \quad A = V\Lambda V^{\mathrm{T}}, \quad \textit{and} \quad V^{\mathrm{T}}U = I.$$

*The diagonals of $\Lambda$ are the eigenvalues of $(A, B)$, and the columns of $U$ are the corresponding eigenvectors.*

*Proof.* The result follows from applying the real spectral theorem to the transformed eigenvalue problem $R^{-\mathrm{T}}AR^{-1}y = \lambda y$. □

**Corollary 1.5.3.1.** *Let $A, B \in \mathbb{R}^{n\times n}$ be symmetric. If there exists a shift $\mu$ such that $A + \mu B$ is positive definite, then the matrix pencil $(A, B)$ has all real eigenvalues.*

It is important to note that the converse of this statement is not necessarily true. As previously mentioned, if the matrix $B$ is a symmetric positive definite matrix, then it induces an inner product, which in turn induces a norm, with a corresponding dual norm induced by $B^{-1}$. If a matrix pencil $(A, B)$ has $A$ symmetric and $B$ symmetric positive definite, the pencil $(A, B)$ is called *symmetric definite*.

**Lemma 1.5.4.** *Let $A \in \mathbb{R}^{n\times n}$ be symmetric, and let $B$ be symmetric positive definite. Let $\lambda_{\max}$ and $\lambda_{\min}$ denote the largest and smallest eigenvalues, in magnitude, of the pencil $(A, B)$. Then*

$$||A||_{B,B^{-1}} = |\lambda_{\max}|$$

*and*

$$\kappa_B(A) = ||A||_{B,B^{-1}}||A^{-1}||_{B^{-1},B} = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}.$$

*Proof.* Let $A = V\Lambda V^{\mathrm{T}}$ be the spectral decomposition of $A$ with respect to the $B$-inner product, and let $U$ be defined by $BU = V$. Then

$$||A||_{B,B^{-1}} = \max_{x\neq 0} \frac{||Ax||_{B^{-1}}}{||x||_B} = \max_{x\neq 0} \frac{\sqrt{x^{\mathrm{T}}AB^{-1}Ax}}{\sqrt{x^{\mathrm{T}}Bx}}$$

can be written as

$$\begin{aligned}||A||_{B,B^{-1}} &= \max_{x\neq 0} \frac{\sqrt{x^{\mathrm{T}}V\Lambda V^{\mathrm{T}}B^{-1}V\Lambda V^{\mathrm{T}}x}}{\sqrt{x^{\mathrm{T}}Bx}} \\ &= \max_{u\neq 0} \frac{\sqrt{u^{\mathrm{T}}U^{\mathrm{T}}V\Lambda V^{\mathrm{T}}B^{-1}V\Lambda V^{\mathrm{T}}Uu}}{\sqrt{u^{\mathrm{T}}UBUu}} \\ &= \max_{u\neq 0} \frac{\sqrt{u^{\mathrm{T}}\Lambda^2 u}}{\sqrt{u^{\mathrm{T}}u}} = |\lambda_{\max}|\end{aligned}$$

Similarly,

$$||A^{-1}||_{B^{-1},B} = \max_{y\neq 0} \frac{\sqrt{y^{\mathrm{T}}A^{-1}BA^{-1}y}}{\sqrt{y^{\mathrm{T}}B^{-1}y}} = \max_{v\neq 0} \frac{\sqrt{v^{\mathrm{T}}\Lambda^{-2}v}}{v^{\mathrm{T}}v} = \frac{1}{|\lambda_{\min}|}.$$

□

Given a symmetric matrix $A$ and a vector $x$, the *Rayleigh-quotient* of $x$ is defined to be $x^{\mathrm{T}}Ax/x^{\mathrm{T}}x$. This quotient typically appears in estimations of eigenvalues, as if $x$ is an eigenvector of $A$ with corresponding eigenvalue $\lambda$, then $\lambda = x^{\mathrm{T}}Ax/x^{\mathrm{T}}x$. The Rayleigh-quotient can be generalized to the generalized eigenvalue problem with respect to symmetric definite matrix pencil.

**Definition 1.5.2.** Let $(A, B)$ be a matrix pencil in which $A$ is symmetric, and $B$ is a symmetric positive definite matrix. Given a vector $x$, the Rayleigh quotient is given by $x^{\mathrm{T}}Ax/x^{\mathrm{T}}Bx$.

**Theorem 1.5.5.** *Let $(A, B)$ be a matrix pencil in which $A$ is symmetric, and $B$ is symmetric positive definite. Let $\lambda_{\max}$ denote the largest generalized eigenvalue and $\lambda_{\min}$ the smallest generalized eigenvalue. Then*

$$\lambda_{\max} = \max_{x\neq 0} \frac{x^{\mathrm{T}}Ax}{x^{\mathrm{T}}Bx}, \quad \textit{and} \quad \lambda_{\min} = \min_{x\neq 0} \frac{x^{\mathrm{T}}Ax}{x^{\mathrm{T}}Bx}.$$

*Proof.* The result follows from applying the equivalent standard eigenvalue result to the matrix $R^{-\mathrm{T}}AR^{-1}$, where $R$ is the Cholesky factor of $B$. □

## 1.6 The Lanczos Process

When dealing with a symmetric matrix $A$, it is often convenient to apply a similarity transform to $A$ to form a tridiagonal matrix and work with the equivalent transformed problem instead of the original problem. Householder transformations are a convenient tool to perform this transformation. Unfortunately, methods involving Householder reflectors do not scale well as the dimension of the matrix $A$ grows. As the dimension gets larger, it becomes more practical to work with an iterative process that builds the tridiagonal matrix one step at a time, as opposed to all at once. The *Lanczos process* is one method to achieve this goal. The Lanczos process is typically presented using the standard inner product $\langle x, y\rangle = y^{\mathrm{T}}x$, however, inner products induced by symmetric positive definite matrices can be used as well.

Let $A, B \in \mathbb{R}^{n\times n}$ be two symmetric matrices. Additionally, assume that $B$ is positive definite. Let $R$ be the Cholesky factor of $B$ so that $B = R^{\mathrm{T}}R$. Define $\bar{A}$ as $R^{-\mathrm{T}}AR^{-1}$. From the theory of Householder reflectors, it is well known that for any vector $\bar{u}_1$ such that $||\bar{u}_1|| = 1$, there exists an orthogonal matrix $\bar{U}$ such that $\bar{U}e_1 = \bar{u}_1$ and $\bar{U}^{\mathrm{T}}\bar{A}\bar{U}$ is a tridiagonal matrix

$$\bar{U}^{\mathrm{T}}\bar{A}\bar{U} = T = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix}.$$

Now, $\bar{U}^{\mathrm{T}}\bar{A}U = \bar{U}^{\mathrm{T}}R^{-\mathrm{T}}AR^{-1}\bar{U}$. Let $U = R^{-1}\bar{U}$, so that $U^{\mathrm{T}}AU = T$. Recall $B$ induces an inner product $\langle x, y\rangle_B = y^{\mathrm{T}}Bx$. The matrix $U$ is $B$-orthogonal, as $U^{\mathrm{T}}BU = \bar{U}^{\mathrm{T}}R^{-\mathrm{T}}BR^{-1}\bar{U} = \bar{U}^{\mathrm{T}}\bar{U} = I$. Thus, for any $B$-orthonormal vector $u_1$, there exists a $B$-orthogonal matrix $U$ such that

$$Ue_1 = u_1, \quad U^{\mathrm{T}}AU = T, \quad \text{and} \quad U^{\mathrm{T}}BU = I.$$

These three equations form the foundation of the Lanczos process. From the second and third equations, it holds that

$$AU = BUT.$$

Let $V = BU$. As $B$ is symmetric positive definite, $B^{-1}$ exists and is also symmetric positive definite, and $V$ is $B^{-1}$-orthogonal. In terms of the columns of $U$ and $V$, the above equation becomes

$$\begin{pmatrix} Au_1 & Au_2 & Au_3 & \cdots & Au_n \end{pmatrix} = \begin{pmatrix} v_1 & v_2 & v_3 & \cdots & v_n \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix}.$$

Thus,

$$\begin{aligned} Au_1 &= \alpha_1 v_1 + \beta_2 v_2, \text{ and} \\ Au_i &= \beta_i v_{i-1} + \alpha_i v_i + \beta_{i+1} v_{i+1} \text{ for all } i = 2, \ldots, n, \end{aligned}$$

with $\beta_{n+1} = 0$. As $U^{\mathrm{T}}V = I$, $u_i^{\mathrm{T}}v_j = 0$ if $i \neq j$, and 1 if $i = j$. The scalars $\alpha_i$ are then simply $u_i^{\mathrm{T}}Au_i$.

Setting $\beta_1 = 0$, it holds that

$$\begin{aligned}\beta_{i+1} v_{i+1} =& A u_i - \beta_i v_{i-1} - \alpha_i v_i, \quad \text{and} \\ \beta_{i+1} u_{i+1} =& B^{-1} A u_i - \beta_i u_{i-1} - \alpha_i u_i.\end{aligned}$$

Thus,

$$\begin{aligned}\beta_{i+1} &= \sqrt{(A u_i - \beta_i v_{i-1} - \alpha_i v_i)^{\mathrm{T}} B^{-1} (A v_i - \beta_i v_{i-1} - \alpha_i v_i)}, \\ v_{i+1} &= \frac{1}{\beta_{i+1}} (A u_i - \beta_i v_{i-1} - \alpha_i v_i), \text{ and} \\ u_{i+1} &= \frac{1}{\beta_{i+1}} (B^{-1} A u_i - \beta_i u_{i-1} - \alpha_i u_i).\end{aligned}$$

These steps constitute the main loop of the Lanczos process. They can be summarized with the following steps:

1. $w = A u_i - \beta_i v_{i-1}$.
2. $\alpha_i = u_i^{\mathrm{T}} w$.
3. $w \leftarrow w - \alpha_i v_i$.
4. $\bar{u}_{i+1} = B^{-1} w$.
5. $\beta_{i+1} = \sqrt{w^{\mathrm{T}} \bar{u}_{i+1}}$.
6. $v_{i+1} = \frac{1}{\beta_{i+1}} w$.
7. $u_{i+1} = \frac{1}{\beta_{i+1}} \bar{u}_{i+1}$.

In a practical implementation of these steps, the vectors $\bar{u}_j$ and $u_j$ do not need to be stored separately. Let $U_i$ and $V_i$ be the $n \times i$ dimensional matrices whose columns consist of the vectors $u_1, \ldots, u_i$ and $v_1, \ldots, v_i$, respectively. Let $T_i$ be the upper left $i \times i$ block of $T$. Then

$$A U_i = V_i T_i + \beta_{i+1} v_{i+1} e_i^{\mathrm{T}} = V_{i+1} \widehat{T}_i, \tag{1.1}$$

where

$$\widehat{T}_i = \begin{pmatrix} T_i \\ \beta_{i+1} e_i^{\mathrm{T}} \end{pmatrix}.$$

Equation (1.1) is referred to as the Lanczos decomposition, and is the foundation of numerous iterative algorithms for solving both linear systems and eigenvector problems. Note that

$$\mathrm{range}(U_k) = \mathrm{Span}\{u_1, B^{-1} A u_1, \ldots, (B^{-1} A)^{k-1} u_1\} = \mathcal{K}_k(B^{-1} A, u_1),$$

the $k$-th *Krylov subspace* of $B^{-1}A$ and $u_1$.

Consider the case where at the $k$-th iteration, it is determined that $\beta_{k+1} = 0$. Then, by (1.1),

$$AU_k = V_k T_k = BU_k T_k.$$

As $T_k$ is symmetric, there exists a basis of $\mathbb{R}^k$ consisting of orthonormal eigenvectors of $T_k$. Let $(z, \lambda)$ be some eigenpair of $T_k$. Then

$$AU_k z = BU_k T_k z = \lambda BU_k z.$$

The vector $q = U_k z$ solves the generalized eigenvector problem $Aq = \lambda Bq$. Let $Z$ be the matrix consisting of all orthonormal eigenvectors of $T_k$, and $Q = U_k Z$. Then each column of $Q$ is a generalized eigenvector of the matrix pencil $(A, B)$, and the columns of $Q$ form a basis for an eigenspace of $(A, B)$. In this case, the Lanczos process has exhausted an eigenspace of $(A, B)$, and the Lanczos process has broken down. This occurs when the initial vector $u_1 \in \text{range}(Q)$. In most applications of the Lanczos process, this indicates that, in exact arithmetic, the solution to the problem has been found. However, this is not always the case. Sometimes, the Lanczos process needs to be continued. In that case, a new vector $u_{k+1}$ is chosen so that $u_{k+1}^{\mathrm{T}} B u_{k+1} = 1$, and $U_k^{\mathrm{T}} B u_{k+1} = 0$. Typical implementations of the Lanczos process only store the matrix $T$ in memory, not the matrices $U$ and $V$. Thus, in the case of breakdown, the Lanczos process would need to be rerun from the beginning in order to ensure that the chosen $u_{k+1}$ does not have any components in the exhausted subspace.

### 1.6.1 Lanczos-CG

The Lanczos process provides a simple, powerful method for transforming certain linear algebra problems into equivalent problems which are significantly easier to solve. For instance, consider solving the system of linear equations $Ax = b$ when $A$ is symmetric positive definite. Let $B$ be a symmetric positive definite matrix such that $B \approx A$, and $Bu = v$ is straightforward to solve. Popular choices include $B = \text{diag}(A)$, the matrix consisting of only the diagonals of $A$, or the incomplete Cholesky decomposition of $A$ (see [24]). Suppose the Lanczos process is initialized with $v_1 = b/\beta_1$, where $\beta_1$ is a constant so that $(v_1 B^{-1} v_1)^{1/2} = 1$, and $\beta_1 BUe_1 = b$. The idea is to find, at each iteration, an approximate solution $x_k \in \text{range}(U_k)$ such that the residual vector $r_k = b - Ax_k$ satisfies

$$r_k \perp \text{range}(U_k) = \mathcal{K}_k(B^{-1}A, B^{-1}b).$$

Thus, at each iteration, the following projected subproblem needs to be solved:

$$U_k^{\mathrm{T}} A U_k y_k = T_k y_k = U_k^{\mathrm{T}} b = \beta_1 e_1.$$

The matrix $A$ is positive definite, and therefore each $T_k$ is positive definite, so $T_k$ has a square root free Cholesky decomposition $T_k = L_k D_k L_k^{\mathrm{T}}$. This decomposition is the symmetric indefinite factorization of $T_k$, where the positive definiteness of $T_k$ guarantees that $D_k$ is positive definite and diagonal. So,

$$\begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_k & 1 \end{pmatrix} \begin{pmatrix} d_1 & & & & \\ & d_2 & & & \\ & & d_3 & & \\ & & & \ddots & \\ & & & & d_k \end{pmatrix} \begin{pmatrix} 1 & l_2 & & & \\ & 1 & l_3 & & \\ & & 1 & \ddots & \\ & & & \ddots & l_k \\ & & & & 1 \end{pmatrix}.$$

Therefore,

$$\begin{aligned} d_1 &= \alpha_1, \\ l_k &= \beta_k / d_k, \text{ and} \\ d_k &= \alpha_k - l_k d_{k-1} l_k \quad \text{for all} \quad k = 2, \ldots, n. \end{aligned}$$

As new entries are added to $T$, the previous Cholesky factors remain the same. These factors can be stored in memory, so that the full Cholesky decomposition of $T_k$ need not be recomputed at each iteration. Now, introduce a new vector $z_k \in \mathbb{R}^k$ such that $D_k L_k^{\mathrm{T}} y_k = z_k$. Then $L_k z_k = \beta_1 e_1$, or

$$\begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_k & 1 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_k \end{pmatrix} = \begin{pmatrix} \beta_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Then $z_k$ is easily solved to be

$$z_k = \begin{pmatrix} \beta_1 \\ -l_2\xi_1 \\ -l_3\xi_2 \\ \vdots \\ -l_k\xi_{k-1} \end{pmatrix} = \begin{pmatrix} z_{k-1} \\ -l_k\xi_{k-1} \end{pmatrix}.$$

The vector $z_k$ need not be fully computed at each iteration. Only the last entry of $z_k$ is unknown at iteration $k$. Define the matrix $P_k$ by $U_k = P_k D_k L_k^{\mathrm{T}}$. Then

$$x_k = U_k y_k = P_k D_k L_k^{\mathrm{T}} y_k = P_k z_k.$$

Let $p_1, \ldots, p_k$ denote the columns of $P_k$. Then

$$\begin{pmatrix} u_1 & u_2 & \cdots & u_k \end{pmatrix} = \begin{pmatrix} p_1 & p_2 & \cdots & p_k \end{pmatrix} \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_k \end{pmatrix} \begin{pmatrix} 1 & l_2 & & \\ & 1 & l_3 & \\ & & \ddots & l_k \\ & & & 1 \end{pmatrix},$$

so that

$$p_1 = (1/d_1)u_1 \quad \text{and} \quad p_k = (1/d_k)(u_k - d_{k-1} l_k p_{k-1}).$$

Notice that at each iteration, the first $k-1$ columns of $P_k$ are simply the columns of $P_{k-1}$. Thus,

$$x_k = P_k z_k = P_{k-1} z_{k-1} + p_k \xi_k = x_{k-1} + p_k \xi_k.$$

Now, consider the residual vector $r_k = b - Ax_k$. It holds that

$$r_k = b - Ax_k = b - AU_k y_k = \beta_1 V_k e_1 - (V_k T_k + \beta_{k+1} v_{k+1} e_k^{\mathrm{T}}) y_k,$$

or

$$r_k = V_k(\beta_1 e_1 - T_k y_k) - \beta_{k+1} v_{k+1} e_k^{\mathrm{T}} y_k = -\beta_{k+1} v_{k+1} e_k^{\mathrm{T}} y_k.$$

This shows that the magnitude of the residual at iteration $k$ depends only on $\beta_{k+1}$ and the last entry of $y_k$. This is how the method can converge before the Lanczos process exhausts a complete eigenspace.

Now, in the $B^{-1}$ norm, this becomes

$$r_k^{\mathrm{T}} B^{-1} r_k = \beta_{k+1}^2 (e_k^{\mathrm{T}} y_k)^2.$$

From the definitions of $y_k$, $\xi_k$, $l_k$, and $d_k$, it holds that $r_k^{\mathrm{T}} B^{-1} r_k = \xi_{k+1}^2$, giving a convenient way to measure the convergence of the algorithm.

It turns out that this algorithm is, in exact precision, equivalent to the more well-known preconditioned conjugate gradient algorithm (PCG). The PCG algorithm is presented in Algorithm 1.1 for completeness.

**Algorithm 1.1.** Preconditioned Conjugate Gradient Algorithm

Given $A \succ 0$, $B \approx A$ such that $B \succ 0$ and $Bu = v$ is simple to compute, $x_0 \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $\epsilon > 0$.
Returns $x$ such that $||Ax - b||_2^2 \leq \epsilon$.
$k \leftarrow 0$.
$x \leftarrow x_0$.
$r_0 \leftarrow b - Ax_0$.
**if** $||r_0||_2^2 \leq \epsilon$ **then**
    exit.
**end if**
Solve $Bz_0 = r_0$.
$p_0 \leftarrow z_0$.
**while** Not Converged **do**
    $\alpha_k \leftarrow \frac{r_k^{\mathrm{T}} z_k}{p_k^{\mathrm{T}} A p_k}$
    $x \leftarrow x + \alpha_k p_k$
    $r_{k+1} \leftarrow r_k - \alpha_k A p_k$
    **if** $||r_{k+1}||_2^2 \leq \epsilon$ **then**
        exit.
    **end if**
    Solve $Bz_k = r_k$.
    $\beta_{k+1} \leftarrow \frac{r_{k+1}^{\mathrm{T}} z_{k+1}}{r_k^{\mathrm{T}} z_k}$
    $p_{k+1} = z_{k+1} + \beta_{k+1} p_k$
**end while**

Note that these parameters $\alpha_j$ and $\beta_j$ are not identical to the parameters appearing in the Lanczos process. While Algorithm 1.1 is presented using the most commonly used notation, a change of variables is used to avoid confusion. Let $\gamma_i$ denote the PCG parameter $\alpha_i$, and $\theta_i$ denote the parameter $\beta_i$. The residual in Algorithm 1.1 is updated at each iteration using the previous residual and the product of the matrix $A$ with the current search direction $p_k$. Eliminating the search direction $p_k$ from this update yields the three term-recursion

$$r_{k+1} = -\gamma_k A B^{-1} r_k + (1 - \frac{\gamma_k \theta_k}{\gamma_{k-1}}) r_k + \frac{\gamma_k \theta_k}{\gamma_{k-1}} r_{k-1}.$$

Compare this with the three-term recursion in the Lanczos process for $v_{i+1}$:

$$v_{i+1} = \frac{1}{\beta_{i+1}} AB^{-1}v_i - \frac{\alpha_i}{\beta_{i+1}} v_i - \frac{\beta_i}{\beta_{i+1}} v_{i-1}.$$

Recall that the vectors $v_i$ have unit $B^{-1}$ norm. Therefore,

$$v_{i+1} = \frac{(-1)^i}{||r_{i+1}||_{B^{-1}}} r_{i+1}.$$

Thus, the search directions utilized in each method are equivalent to each other. Furthermore, considering that both algorithms make the optimal choice of step length at each iteration, the two must be identical.

**Theorem 1.6.1** (PCG Convergence)**.** *Let $A$, $B \in \mathbb{R}^{n\times n}$ be two symmetric positive definite matrices, where $B^{-1} \approx A$ is such that solving systems of the form $Bu = v$ is straightforward, and let $b \in \mathbb{R}^n$. Suppose that Algorithm 1.1, or equivalently, Lanczos-CG, is used to solve the system $Ax = b$ using some initial approximation $x_0$. Let $x^*$ be the exact solution, $x_k$ the $k$-th iterate, and $e_k = x^* - x_k$. Then*

$$||e_k||_A \leq 2\left(\frac{\sqrt{\kappa_B(A)} - 1}{\sqrt{\kappa_B(A)} + 1}\right)^k ||e_0||_A,$$

*where $\kappa_B(A) = \lambda_{\max}(A, B)/\lambda_{\min}(A, B)$.*

The above result shows that the worst-case convergence rate of Lanczos-CG depends on the square root of the condition number of the matrix $A$ in the $B$-norm. In practice, Lanczos-CG behaves exceptionally well when a strong preconditioner is used and performs much faster than other methods.

In practical cases, PCG is preferred over Lanczos-CG due to the fact that one less vector needs to be stored. However, this example demonstrates the power of the Lanczos process and how it can be leveraged to find an approximate solution for various problems. For instance, a similar procedure can be utilized to solve generalized eigenvalue problems of the form $Ax = \lambda Bx$ by finding the eigenvalues of $T_k$ at each iteration. Suppose the LQ decomposition of $\widehat{T}_k$ is taken at each iteration instead of the Cholesky decomposition of $T_k$. In that case, one can derive the popular MINRES algorithm for solving equations $Ax = b$ when $A$ is only assumed to be symmetric.

It is crucial to note that the Lanczos process has drawbacks. In his Ph.D. thesis [26], Paige demonstrated explicit bounds on the errors of the Lanczos vectors as the algorithm proceeds in finite-precision arithmetic. Unfortunately, as the algorithm proceeds, the round-off errors build up until, eventually, the vectors lose their $B$-orthogonality. Many schemes for correcting this issue have been proposed, such as the

partial reorthogonalization scheme presented in [27]. Unfortunately, reorthogonalization schemes require either storing the Lanczos vectors or rerunning the Lanczos process from the beginning once a loss of orthogonality has been detected.

One alternative to reorthogonalization is restarting. For example, most practical implementations of Algorithm 1.1 include a restart technique so that the Lanczos process is restarted before any significant errors can accumulate. However, restarting schemes are typically algorithm specific and are not discussed here.

### 1.6.2 The Block Lanczos Process

The Lanczos process need not be initialized with a single vector $u_1$. For example, consider the problem of solving the system $AX = C$, where $A \in \mathbb{R}^{n\times n}$ is positive definite, $C \in \mathbb{R}^{n\times m}$, and $X \in \mathbb{R}^{n\times m}$. Lanczos-CG would need to be applied $m$ times to solve this problem. Alternatively, the Lanczos process can be formulated to use $n \times m$ blocks of vectors to avoid applying the same process multiple times. In what follows, assume without loss of generality that $m$ divides $n$.

Let $\widehat{U}_1 = (u_{1,1}, u_{1,2}, \ldots, u_{1,m})$ denote an $n \times m$ matrix with $B$-orthonormal columns, and let $\widehat{V}_1 = B\widehat{U}_1$. Then there exists a $B$-orthogonal matrix $U$ and a $B^{-1}$-orthogonal matrix $V$ whose first $m$ columns are $\widehat{U}_1$ and $\widehat{V}_1$, respectively, such that $U^{\mathrm{T}}AU$ is a block tridiagonal matrix $T$. It still holds then that $AU = VT$. Let $\widehat{U}_j$ and $\widehat{V}_j$ denote the $j$-th block of $m$ columns of the matrices $U$ and $V$. Equating the columns of $AU = VT$ yields

$$\begin{pmatrix} A\widehat{U}_1 & A\widehat{U}_2 & \cdots & A\widehat{U}_{n/m} \end{pmatrix} = \begin{pmatrix} \widehat{V}_1 & \widehat{V}_2 & \cdots & \widehat{V}_{n/m} \end{pmatrix} \begin{pmatrix} A_1 & B_2^{\mathrm{T}} & & \\ B_2 & A_2 & & \\ & \ddots & \ddots & B_{n/m}^{\mathrm{T}} \\ & & B_{n/m} & A_{n/m} \end{pmatrix},$$

where $A_i$ and $B_i \in \mathbb{R}^{m\times m}$, and each $A_i$ is symmetric, for all indices $i$. Thus,

$$\begin{aligned} A\widehat{U}_1 &= \widehat{V}_1 A_1 + \widehat{V}_2 B_2, \quad \text{and} \\ A\widehat{U}_i &= \widehat{V}_{i-1} B_i^{\mathrm{T}} + \widehat{V}_i A_i + \widehat{V}_{i+1} B_{i+1} \quad \text{for all } i = 2, \ldots, n, \end{aligned}$$

with $B_{n/m+1} = 0$. As before, it holds that $A_i = \widehat{U}_i^{\mathrm{T}} A \widehat{U}_i$, and

$$\begin{aligned} \widehat{V}_{i+1} B_{i+1} &= A\widehat{U}_i - \widehat{V}_{i-1} B_i^{\mathrm{T}} - \widehat{V}_i A_i, \\ \widehat{U}_{i+1} &= B^{-1} \widehat{V}_{i+1}, \quad \text{and} \\ \widehat{U}_{i+1}^{\mathrm{T}} \widehat{V}_{i+1} &= I. \end{aligned}$$

Unlike the $m = 1$ case, there is some freedom in choosing how to form $B_{i+1}$. The most straightforward method for forming $B_{i+1}$ is to perform Gram-Schmidt biorthogonalization on the columns of the matrices $A\widehat{U}_i - \widehat{V}_{i-1} B_i^{\mathrm{T}} - \widehat{V}_i A_i$ and $B^{-1}(A\widehat{U}_i - \widehat{V}_{i-1} B_i^{\mathrm{T}} - \widehat{V}_i A_i)$. This yields a matrix $B_{i+1}$ that is upper triangular, in which case $T_{i+1}$ is not only a block triangular matrix but a banded matrix. Column-pivoting can be included to better detect when $B_{i+1}$ becomes rank-deficient, although this destroys the banded structure of $T$. A block version of Lanczos-CG can be derived by taking a block square root free Cholesky decomposition of $T_k$ at each iteration and forming the appropriate approximate solution to the matrix $X$. The block-Lanczos process is most commonly used for finding multiple eigenpairs simultaneously. It shall be used in the shifted and inverted GLTR algorithm to allow convergence in the hard case and to enable the warm-starting of the algorithm.

## 1.7 Other Results

Farkas' lemma is a crucial result used when proving the optimality conditions for inequality-constrained optimization problems. It has several equivalent statements, all of which are detailed here.

**Lemma 1.7.1** (Farkas' Lemma, [14] Section 7.7)**.** *Let $A \in \mathbb{R}^{n \times k}$ be a nonzero matrix and $c \in \mathbb{R}^n$. Then*

1. *$c^{\mathrm{T}} p \geq 0$ for all $p$ such that $Ap \geq 0$ if and only if $c = A^{\mathrm{T}} y$ for some $y \geq 0$.*
2. *There exists a $p$ such that $c^{\mathrm{T}} p < 0$ and $Ap \geq 0$ if and only if $c \neq A^{\mathrm{T}} y$.*
3. *Exactly one of the following holds:*
   (a) *$c$ can be written as a nonnegative linear combination of the columns of $A^{\mathrm{T}}$.*
   (b) *There exists a vector $p$ such that $c^{\mathrm{T}} p < 0$ and $Ap \geq 0$.*

**Lemma 1.7.2** (Debreu's Lemma, [8])**.** *Given an $m \times n$ matrix $A$ and an $n \times n$ matrix $H$, then $x^{\mathrm{T}} H x > 0$ for all $x \in \mathrm{null}(A)$ if and only if there exists a finite $\bar{\rho} \geq 0$ such that $H + \rho A^{\mathrm{T}} A$ is positive definite for all $\rho > \bar{\rho}$.*

# Chapter 2

# Unconstrained Optimization

## 2.1 Introduction

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{2.1}$$

where $x$ is a real vector with $n$ components, and $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous function. $f$ may be assumed to be continuously differentiable, or twice continuously differentiable, depending on the method being examined. Generally, the goal would be to find a point $x^*$ that is a *global minimizer*, i.e., a point such that

$$f(x^*) \leq f(x) \text{ for all } x \in \mathbb{R}^n. \tag{2.2}$$

Unfortunately, this goal is beyond the scope of what is feasible. This is due to the fact that optimization algorithms generally only have access to *local* information about the function $f$, so instead, practical methods are developed with the goal of finding *local minimizers*.

**Definition 2.1.1** (Local Unconstrained Minimizer, [16] Chapter 3)**.** A point $x^*$ is a *local unconstrained minimizer* to problem (2.1) if there exists a neighborhood $\mathcal{B}$ of $x^*$ such that

$$f(x^*) \leq f(x) \text{ for all } x \in \mathcal{B}.$$

This is also sometimes referred to as a *weak local unconstrained minimizer* in order to distinguish it from a *strict local unconstrained minimizer*.

**Definition 2.1.2** (Strict Local Unconstrained Minimizer, [16] Chapter 3)**.** A local unconstrained minimizer

$x^*$ is a *strict local unconstrained minimizer* if there exists a neighborhood $\mathcal{B}$ of $x^*$ such that

$$f(x^*) < f(x) \text{ for all } x \in \mathcal{B}, x \neq x^*.$$

Strict local minimizers may have other local minimizers that are arbitrarily close. A stronger definition is that of an *isolated local minimizer*.

**Definition 2.1.3** (Isolated Local Unconstrained Minimizer, [16] Chapter 3)**.** A local unconstrained minimizer $x^*$ is an *isolated local unconstrained minimizer* if there exists an open ball $\mathcal{B}(x^*, \delta)$ about $x^*$ of radius $\delta > 0$, such that $x^*$ is the only unconstrained minimizer in $\mathcal{B}(x^*, \delta)$.

**Theorem 2.1.1** (Isolated minimizers are strict, [16] Chapter 3)**.** *Given $f : \mathbb{R}^n \to \mathbb{R}$, an isolated unconstrained minimizer $x^*$ of $f$ is a strict unconstrained minimizer.*

*Proof.* The result is proved using a contrapositive argument. Let $x^*$ be a local minimizer of $f$ that is not strict. Then there exists a $\delta > 0$ such that

$$f(x^*) \leq f(x) \quad \text{for all } x \text{ such that } ||x - x^*|| < \delta.$$

As $x^*$ is not strict, there exists a point $\widehat{x} \neq x^*$ such that $||\widehat{x} - x^*|| < \frac{1}{3}\delta$ and $f(\widehat{x}) = f(x^*)$. Then $\mathcal{B}(\widehat{x}, \frac{1}{2}\delta) \subset \mathcal{B}(x^*, \delta)$. This implies that

$$f(\widehat{x}) \leq f(x) \text{ for all } x \in \mathcal{B}(\widehat{x}, \tfrac{1}{2}\delta).$$

Thus, $\widehat{x}$ is another local unconstrained minimizer of $f$ within $\mathcal{B}(x^*, \delta)$. The radius $\delta$ is arbitrary and can be taken to be as small as necessary, so $x^*$ is not isolated. Therefore, $x^*$ can only be an isolated minimizer if it is also a strict minimizer. □

## 2.2 Optimality Conditions

The definitions of minimizers given so far are not particularly useful when it comes to determining if a point is optimal because they require checking infinitely many points in a neighborhood of a potential optimal solution. To formulate practical algorithms, a method is needed to characterize minimizers without having to check nearby points. Fortunately, if $f$ is differentiable, then the derivative at a point gives local information about $f$.

**Theorem 2.2.1** (First-order necessary condition for an unconstrained local minimizer, [16] Chapter 3)**.** *Given $f : \mathbb{R}^n \to \mathbb{R}$, let $x^*$ be a local unconstrained minimizer of $f$. Assume that $f$ is differentiable at $x^*$. Then $\nabla f(x^*) = 0$.*

*Proof.* By definition (2.1.1),

$$f(x^*) \leq f(x^* + \alpha p) \text{ for all } p \in \mathbb{R}^n \text{ and, given } p, \text{ all sufficiently small } \alpha. \tag{2.3}$$

As $f$ is differentiable at $x^*$, it holds that if $\alpha \downarrow 0$, then

$$\lim_{\alpha \to 0+} \frac{1}{\alpha}(f(x^* + \alpha p) - f(x^*)) = \nabla f(x^*)^{\mathrm{T}} p \text{ for all } \quad p \in \mathbb{R}^n. \tag{2.4}$$

Combining (2.3) and (2.4) shows that $\nabla f(x^*)^{\mathrm{T}} p \geq 0$ for all $p \in \mathbb{R}^n$, thus $\nabla f(x^*) = 0$. □

The proof can be applied under the weaker condition that $f$ is only Gateaux-differentiable at $x^*$. Points that satisfy $\nabla f(x) = 0$ are referred to as both stationary points and as critical points. It is crucial to note that being a stationary point is not a sufficient condition for being an unconstrained minimizer. Consider the functions $f(x) = \frac{1}{2}x^2$ and $g(x) = \frac{1}{6}x^3$. The point $x = 0$ satisfies $\nabla f(x) = 0$ and $\nabla g(x) = 0$, but is only a minimizer for the function $f$. Thus, any method that naively attempts to solve for $\nabla f(x) = 0$ may converge to an arbitrary stationary point. Therefore, stricter conditions are needed.

**Theorem 2.2.2** (Second-order necessary conditions for an unconstrained minimizer, [16] Chapter 3)**.** *Given $f : \mathbb{R}^n \to \mathbb{R}$, let $x^*$ be a local unconstrained minimizer of $f$. Assume that $f$ is twice differentiable at $x^*$. Then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$.*

*Proof.* Theorem 2.2.1 shows that $\nabla f(x^*) = 0$. By the definition of the second derivative,

$$\lim_{\alpha \to 0} \frac{1}{\alpha^2}(f(x^* + \alpha p) - f(x^*)) = \frac{1}{2} p^{\mathrm{T}} \nabla^2 f(x^*) p \text{ for all } p \in \mathbb{R}^n.$$

Suppose that $\nabla^2 f(x^*)$ is not positive semidefinite. Then there exists a vector $q \in \mathbb{R}^n$ such that $q^{\mathrm{T}} \nabla^2 f(x^*) q < 0$. Thus

$$\lim_{\alpha \to 0} \frac{1}{\alpha^2}(f(x^* + \alpha q) - f(x^*)) < 0.$$

This contradicts that $x^*$ is a local minimizer, so $\nabla^2 f(x^*) \succeq 0$. □

Theorem 2.2.2 only provides necessary conditions for optimality. Points that satisfy $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$ are not necessarily local minimizers. Again, consider the function $f(x) = \frac{1}{6}x^3$. This

simple cubic function satisfies $\nabla f(0) = 0$ and $\nabla^2 f(0) = 0$, but does not have zero as a local minimizer. The following theorem proves stricter conditions that guarantee that a point $x$ is a local minimum.

**Theorem 2.2.3** (Second-order sufficient conditions for an unconstrained minimizer, [16] Chapter 3)**.** *Given $f : \mathbb{R}^n \to \mathbb{R}$, and $x^* \in \mathbb{R}^n$, assume that the second derivative of $f$ exists at $x^*$. If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$, then $x^*$ is an isolated (and therefore strict) local unconstrained minimizer of $f$.*

*Proof.* First, it is shown that $x^*$ is a strict minimizer. By assumption, $\nabla f(x^*) = 0$, so

$$\lim_{\alpha \to 0} \frac{1}{\alpha^2}(f(x^* + \alpha p) - f(x^*)) = \frac{1}{2} p^{\mathrm{T}} \nabla^2 f(x^*) p \quad \text{for all } p \in \mathbb{R}^n.$$

As $\nabla^2 f(x^*) \succ 0$, $p^{\mathrm{T}} \nabla^2 f(x^*) p > 0$ for all $p \neq 0$, thus $f(x^* + \alpha p) - f(x^*) > 0$ for all $p \neq 0$ and $\alpha$ sufficiently small. Therefore, there exists a neighborhood about $x^*$ in which $f(x) > f(x^*)$ for all $x \neq x^*$, so that $x^*$ is a strict local minimizer.

Now it is shown that $x^*$ is also an isolated minimizer. As the second derivative of $f$ exists at $x^*$, the gradient $\nabla f(x)$ exists in a neighborhood of $x^*$ and is continuous at $x^*$. Suppose that $x^*$ is not an isolated minimizer. Then, for every neighborhood of $x^*$, there exists a point $x$ in this neighborhood, which is also a stationary point. Therefore, a sequence of stationary points $\{x_k\}_{k=1}^{\infty}$ can be constructed that is convergent to $x^*$. As the second derivative of $f$ exists at $x^*$,

$$\lim_{||p||_2 \to 0} \frac{1}{||p||_2} ||\nabla f(x^* + p) - \nabla f(x^*) - \nabla^2 f(x^*) p||_2 = 0.$$

As $x_k - x^* \to 0$ and $\nabla f(x_k) = \nabla f(x^*) = 0$ for all $k$, this relation holds for each $p_k = x_k - x^*$. For each $k$, let $u_k = p_k / ||p_k||$. Thus,

$$\lim_{k \to \infty} ||\nabla^2 f(x^*) u_k||_2 = 0.$$

However, $\nabla^2 f(x^*)$ is positive definite, and thus is nonsingular, so

$$||\nabla^2 f(x^*) u||_2 \geq \lambda > 0$$

for any unit length vector $u$, where $\lambda$ is the eigenvalue of $\nabla^2 f(x^*)$ of least magnitude. This creates a contradiction, which shows that there exists a neighborhood in which $x^*$ is the only stationary point. The result follows. □

## 2.3 Directions of Decrease

Consider an objective function $f$ defined on $\mathbb{R}^n$ to minimize, and some point $x$ that is not a local minimizer. The optimality conditions discussed in the previous section can help to form methods for finding a local minimizer by constructing a sequence of points starting at $x$ that converge to a local minimizer. Note that every neighborhood of $x$ must contain points such that the value of $f$ is less than $f(x)$. Consequently, there exists a path starting at $x$ that strictly decreases $f$. The most obvious choice for such a path is a ray that emanates from $x$ (although some methods consider nonlinear paths).

**Definition 2.3.1** (Direction of Decrease, [16] Chapter 3)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. Let $x \in \mathbb{R}^n$. A vector $p \in \mathbb{R}^n$ is a *direction of decrease* for $f$ at $x$ if there exists a positive $\widehat{\alpha}$ such that $f(x + \alpha p) < f(x)$ for all $0 < \alpha < \widehat{\alpha}$.

An immediate result of this definition is that if a direction of decrease exists at $x$, then $x$ is not a minimizer.

**Result 2.3.1** (Existence of a direction of decrease, [16] Chapter 3)**.** *Given $f : \mathbb{R}^n \to \mathbb{R}$, assume that $f$ is continuously differentiable on a convex set $\mathcal{D} \subseteq \mathbb{R}^n$, and let $x \in \text{int}(\mathcal{D})$.*

1. *If a vector $p$ satisfies $\nabla f(x)^{\mathrm{T}} p < 0$, then $p$ is a direction of decrease at $x$*
2. *If $f$ has a second derivative at $x$, then any $p$ satisfying $\nabla f(x)^{\mathrm{T}} p = 0$ and $p^{\mathrm{T}} \nabla^2 f(x) p < 0$ is a direction of decrease at $x$.*

*Proof.* Due to the fact that $x$ is an interior point of $\mathcal{D}$ and $\nabla f(x)$ is continuous on $\mathcal{D}$, the inequality $\nabla f(x)^{\mathrm{T}} p < 0$ implies there exists a $\delta > 0$ such that, for all $\alpha \in [0, \delta)$, $x + \alpha p \in \mathcal{D}$ and $\nabla f(x + \alpha p)^{\mathrm{T}} p < 0$. By the mean-value theorem, for every $\alpha \in (0, \delta)$, there exists a $t \in (0, 1)$ such that

$$f(x + \alpha p) - f(x) = \alpha \nabla f(x + t\alpha p)^{\mathrm{T}} p.$$

It follows that $f(x+\alpha p)-f(x) < 0$, which establishes (1). Conversely, any direction satisfying $\nabla f(x)^{\mathrm{T}} p > 0$ cannot be a direction of decrease. Thus, for part (2), it suffices to consider vectors that satisfy $\nabla f(x)^{\mathrm{T}} p = 0$.

Suppose that $f$ has a second derivative at $x$. Let $p$ be a vector satisfying the properties in (2). Then

$$\lim_{\alpha \to 0} \frac{1}{\alpha^2} (f(x + \alpha p) - f(x)) = \frac{1}{2} p^{\mathrm{T}} \nabla^2 f(x) p < 0,$$

so $f(x + \alpha p) - f(x) < 0$ for $\alpha$ sufficiently small, therefore $p$ is a direction of decrease. □

Vectors $p$ that satisfy $\nabla f(x)^{\mathrm{T}} p < 0$ are referred to as *descent directions*, while vectors that satisfy $p^{\mathrm{T}} \nabla^2 f(x) p < 0$ are referred to as *directions of negative curvature*.

## 2.4 Line-search Methods

In this section it is shown how to use a descent direction $p_k$ found by minimizing a strictly convex local model of the objective function to take a suitable step toward a local minimizer. The goal of this method is to generate a sequence of points $\{x_k\}_{k=1}^{\infty}$ such that $\lim_{k\to\infty} \nabla f(x_k) = 0$, where the sequence of points is generated via an update rule of the form

$$x_{k+1} = x_k + \alpha_k p_k,$$

where $p_k$ is a descent direction at $x_k$, and $\alpha_k$ is a suitable step-length. Such methods are commonly referred to as line-search methods, as the descent directions are formed first, and then a suitable step-length is found by sampling the function $f$ along the ray $\{x : x = x_k + \alpha p, \alpha > 0\}$. Two common methods for choosing the search direction $p_k$ are considered: the steepest-descent method and Newton's method.

### 2.4.1 Sufficient decrease conditions

It has been shown that if $p_k$ is a descent direction at $x_k$, there exists a step-length $\widehat{\alpha}$ such that $f(x_k + \alpha p_k) < f(x_k)$ for all $0 < \alpha < \widehat{\alpha}$. A naive approach would be to choose the step-length $\alpha$ as an arbitrarily small value. Unfortunately, this will lead to an algorithm requiring far too many iterations to achieve a sufficiently accurate local minimizer. It is necessary that any line-search method attempt to find a step length that decreases the objective function without the decrease becoming so small that progress is inhibited. This leads to the notion of *sufficient decrease conditions*.

The most extreme approach would be to choose the first minimizer of $f$ along the ray $\{x_k + \alpha p_k : \alpha > 0\}$. This is referred to as an exact line search. Note that a step length $\alpha_k$ chosen via exact line search must satisfy

$$\frac{d}{d\alpha} f(x_k + \alpha p_k) = \nabla f(x_k + \alpha p_k)^{\mathrm{T}} p_k = 0.$$

The update is taken as $x_{k+1} = x_k + \alpha_k p_k$, and has the property that the gradient of $f$ at the new point is orthogonal to the search direction, i.e., no more progress can be made along the given direction $p_k$. In the case that $f$ is a strictly convex quadratic function of the form $f(x) = \frac{1}{2} x^{\mathrm{T}} H x + g^{\mathrm{T}} x$, then, given a search

direction $p_k$, the optimal step length $\alpha_k$ can be computed as

$$\alpha_k = \frac{-\nabla f(x_k)^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} H p_k} = \frac{-(g + Hx_k)^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} H p_k}.$$

However, exact line searches are generally considered inefficient for more complicated objective functions, as minimizing $f$ along a direction $p_k$ is comparable in work to the overall problem of minimizing $f$ itself. So instead, the exact step length is used for theoretical purposes as the "best" step possible.

The convergence analysis of a line-search method depends on the method used to choose the search direction $p_k$ and the method used to choose the step length $\alpha_k$. For now, suppose that a line-search algorithm yields a sequence of points $\{x_k\}$ such that $f(x_{k+1}) < f(x_k)$ for all $k$. Further, assume that the function $f$ is bounded below (otherwise, the optimization problem is not well posed). Then the strictly decreasing sequence $\{f(x_k)\}$ converges to a limit. Thus, given an initial iterate $x_0$, all iterates lie in the level set $\mathcal{L}(f(x_0)) = \{x : f(x) \leq f(x_0)\}$. If this level set is compact, then $\{f(x_k)\}$ is bounded below and must converge. However, this is not enough to guarantee that $f(x_k)$ converges to the value of $f$ at a solution. More conditions are needed to show that the sequence of iterates $x_k$ converges to an optimal point $x^*$. Thus, instead of simply requiring that $f(x_{k+1}) < f(x_k)$, stricter conditions are enforced to ensure that each new iterate achieves a decrease that is sufficient to guarantee progress to a local minimizer.

**Backtracking and the Armijo condition**

At each iterate $x_k$ of a line-search method, a local model of $f$ at $x_k$, denoted $m_k(x)$, is constructed to measure how much improvement a step actually makes. The step length $\alpha_k$ is then chosen so that the actual reduction in the objective function is at least as large as a multiple of the predicted reduction given by the local model function. To be more precise, $\alpha_k$ is chosen so that

$$f(x_k) - f(x_k + \alpha_k p_k) \geq \eta_A (m_k(x_k) - m_k(x_k + \alpha_k p_k)), \tag{2.5}$$

where $\eta_A$ is held constant throughout the algorithm and $0 < \eta_A < 1$.

For the local model function $m_k$ to be a good model, two conditions are required. First, $m(x_k + \alpha_k p_k) < m_k(x_k)$ for all $\alpha$ sufficiently small, so that $p_k$ is a descent direction not only for $f$ but for $m_k$ as well. Furthermore, it is required that

$$\lim_{\alpha \to 0+} \frac{f(x_k) - f(x_k + \alpha p_k)}{m_k(x_k) - m_k(x_k + \alpha p_k)} = 1,$$

so that as $\alpha \to 0$, the local behavior of $m_k$ more closely matches that of $f$. The model $m_k$ is also typically chosen so that $m_k(x_k) = f(x_k)$. This property can be enforced for higher order derivatives as well, i.e., $\nabla^m m_k(x_k) = \nabla^m f(x_k)$ for all $m = 1, \ldots, M$ for some positive integer $M$. Typical values of $M$ do not exceed 2.

For non-differentiable functions, the choice of the model function $m_k$ can be somewhat unintuitive. For differentiable functions, the first-order Taylor series satisfies these conditions quite nicely. Thus,

$$m_k(x) = \ell_k(x) = f(x_k) + \nabla f(x_k)(x - x_k)$$

gives the predicted reduction

$$m_k(x_k) - m_k(x_k + \alpha p_k) = -\alpha \nabla f(x_k)^{\mathrm{T}} p_k.$$

The search direction $p_k$ may also be chosen via a method that requires a local model of $f$. However, the two model functions need not be the same. If $p_k$ is a descent direction, notice that the predicted reduction is always positive. The sufficient decrease condition (2.5) can be written as

$$f(x_k + \alpha p_k) \leq f(x_k) + \alpha \eta_A \nabla f(x_k)^{\mathrm{T}} p_k, \tag{2.6}$$

commonly called the *Armijo condition*. Any line search attempting to satisfy this condition is called an Armijo line search. However, this condition does not enforce the requirement that the step length $\alpha$ is not too small. A sufficiently small $\alpha$ will always satisfy the Armijo condition. Instead, the goal should be to find the largest $\alpha$ such that the Armijo condition holds. This motivates the backtracking line-search algorithm. Notice that the main computational work is done in evaluating $f(x + \alpha p)$ for each choice of $\alpha$.

**Algorithm 2.1.** Backtracking Line-Search

1: Given $k_{max}$, $0 < \eta_A < 1$, $0 < \gamma_C < 1$, $\alpha_0$, $x$, $p$
2: $k \leftarrow 0$
3: $\alpha \leftarrow \alpha_0$
4: **while** $k \leq k_{max}$ and $f(x + \alpha p) > f(x) + \alpha \eta_A \nabla f(x)^{\mathrm{T}} p$ **do**
5: $\quad \alpha \leftarrow \gamma_C \alpha$
6: **end while**

For methods based on computing search directions via Newton's method, $\alpha_0$ is chosen to be 1, whereas steepest-descent methods may use a different initial choice at each step. This simple strategy works quite well in practice. More sophisticated approaches add additional conditions on $\alpha$. For instance, the Wolfe

conditions require that

1. $f(x_k + \alpha p_k) \leq f(x_k) + \alpha \eta_A \nabla f(x_k)^{\mathrm{T}} p_k$,
2. $\nabla f(x_k + \alpha p_k)^{\mathrm{T}} p_k \geq \eta_W \nabla f(x_k)^{\mathrm{T}} p_k$,

where $\eta_A < \eta_W < 1$. The strong Wolfe conditions are slightly stricter in that the requirements become

1. $f(x_k + \alpha p_k) \leq f(x_k) + \alpha \eta_A \nabla f(x_k)^{\mathrm{T}} p_k$,
2. $|\nabla f(x_k + \alpha p_k)^{\mathrm{T}} p_k| \leq \eta_W |\nabla f(x_k)^{\mathrm{T}} p_k|$.

Less used nowadays is also the Goldstein conditions, which require that

$$f(x_k) + (1 - \eta_A) \nabla f(x_k)^{\mathrm{T}} p_k \leq f(x_k + \alpha_k p_k) \leq f(x_k) + \eta_A \nabla f(x_k)^{\mathrm{T}} p_k,$$

where $0 < \eta_A < \frac{1}{2}$. These three sets of conditions prevent the step length $\alpha$ from becoming too small. Line-search methods that attempt to satisfy these stricter conditions often operate via interpolation and bracketing to find an initial interval in which a step length $\alpha$ satisfies the conditions, then iterate to refine the interval down to a point. Methods for solving the Wolfe conditions are particularly complicated both theoretically and practically. Any practical implementation of a Wolfe line search must go to great lengths to compensate for round-off error.

Two of the most widely used methods for choosing the search directions $p_k$ are the steepest-descent method and Newton's method. First, the steepest-descent method is considered. Suppose that at the current iterate $x_k$ of some iterative method, $\nabla f(x_k) \neq 0$. The descent direction condition states that to proceed, finding a vector such that $\nabla f(x_k)^{\mathrm{T}} p_k < 0$ suffices. Thus, an intuitive choice would be to find a search-direction $p$ that minimizes the quantity $\nabla f(x_k)^{\mathrm{T}} p_k$. Let $g_k = \nabla f(x_k)$. Then the task is to find a vector $p_k$ such that

$$p_k = \underset{p \in \mathbb{R}^n}{\operatorname{argmin}} - g_k^{\mathrm{T}} p. \tag{2.7}$$

Problem (2.7) is unbounded below. However, note that $p_k$ is a search direction, i.e. an iterative method is allowed to search along the open ray $\{x_k + \alpha p_k : \alpha > 0\}$. Thus, the scaling of the search direction does not matter here. Instead, $p_k$ is chosen to be

$$p_k = \underset{p \in \mathbb{R}^n}{\operatorname{argmin}} \{-g_k^{\mathrm{T}} p : ||p|| = \delta\} \tag{2.8}$$

for some scalar $\delta$ and some norm. Then $p_k$ is the vector for which the inequality

$$-||p_k||\ ||g_k||_\star = -\delta||g_k||_\star \leq g_k^{\mathrm{T}} p_k$$

is satisfied with equality. Suppose the norm in question is the Euclidean norm. Then $p_k = -\frac{\delta}{||g_k||_2} g_k$. A straightforward choice for $\delta$ is simply $||g_k||_2$, thus giving the *gradient-descent* direction $p_k = -g_k$. Of course, the Euclidean norm is not the only norm possible. Choosing $\delta$ in (2.8) to be $||g||_\star$ gives the steepest-descent direction in an arbitrary norm. Let $B_k$ be a symmetric positive definite matrix, and let $||\cdot||_{B_k}$ be the norm induced by $B_k$. Then the steepest-descent direction in the $B_k$ norm is given by $p_k = -B_k^{-1} g$. Other common choices are the 1-norm and the $\infty$-norm.

The next common method for choosing a search direction considered is Newton's method. First, consider the steepest-descent direction in the $B_k$-norm, $p_k = -B_k^{-1} g$. This vector can also be found by solving an unconstrained quadratic optimization problem

$$\min_{p \in \mathbb{R}^n} q_k(x_k + p) = f(x_k) + g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} B_k p. \tag{2.9}$$

The function $q_k(x)$ is a *quadratic model* of $f$ at $x_k$, where the Hessian matrix of $f$ at $x_k$ is approximated by a positive definite matrix $B_k$. Suppose now that $f$ is strongly convex and twice-differentiable, i.e., there exists an $m > 0$ such that

$$mI \preceq \nabla^2 f(x) \tag{2.10}$$

for all $x \in \mathcal{L}(f(x_0))$. Then, at each iterate $x_k$, the approximate quadratic model $q_k$ can be taken to be the true quadratic model

$$q_k(x) = f(x_k) + g_k^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}} \nabla^2 f(x_k)(x - x_k).$$

Minimizing the function $Q_k(p) = q_k(x_k + p)$ yields the *Newton direction* $p_k = -\nabla^2 f(x_k)^{-1} g_k$. Thus, the Newton direction is the steepest-descent direction in the $\nabla^2 f(x_k)$ norm.

The convergence behavior of a backtracking line-search method can now be analyzed. First, a theorem is presented that shows that backtracking produces a sufficient decrease regardless of how the search direction is chosen. Next, two theorems are presented, demonstrating convergence when the search direction is chosen via steepest descent and Newton's method.

**Theorem 2.4.1** (An Armijo line-search gives a sufficient decrease, [16] Chapter 3)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be*

*twice continuously differentiable on an open convex set $\mathcal{D} \subseteq \mathbb{R}^n$. Let $x_0 \in \mathcal{D}$ be chosen so that $\mathcal{L}(f(x_0))$ is compact. At each iteration $k$, let $p_k$ be a descent direction such that $||p_k|| \leq \gamma$ for some $\gamma$ independent of $k$, i.e. the sequence $\{||p_k||\}$ is bounded above by $\gamma$. Then*

$$\lim_{k \to 0} |\nabla f(x_k)^{\mathrm{T}} p_k| = 0.$$

*Proof.* The Armijo condition implies

$$f(x_k) - f(x_{k+1}) \geq -\eta_A \alpha_k \nabla f(x_k)^{\mathrm{T}} p_k = \eta_A \alpha_k |\nabla f(x_k)^{\mathrm{T}} p_k|.$$

Thus, the sequence $\{x_k\}$ is well-defined and lies entirely in $\mathcal{L}(f(x_0))$. As $f$ is bounded below on $\mathcal{L}(f(x_0))$, $\{f(x_k)\}$ is a bounded, strictly decreasing sequence and thus converges.

To establish a contradiction, assume that $|\nabla f(x_k)^{\mathrm{T}} p_k|$ does not converge to zero. Thus, there exists an $\varepsilon > 0$ such that $|\nabla f(x_k)^{\mathrm{T}} p_k| \geq \varepsilon$ infinitely many times. Let $\mathcal{G}$ denote the subsequence $\{k : |\nabla f(x_k)^{\mathrm{T}} p_k| \geq \varepsilon\}$. Now, the step length for the backtracking line search is given by $\alpha_k = \gamma_C^{j_k}$, where $j_k$ is the smallest nonnegative integer such that the Armijo conditions holds. Due to the fact that $f(x_k)$ converges, $f(x_k) - f(x_{k+1}) = f(x_k) - f(x_k + \alpha_k p_k) \to 0$, and $|\nabla f(x_k)^{\mathrm{T}} p_k| \geq \varepsilon$, it must hold that $\alpha_k \to 0$. By assumption, $\{||p_k||\}$ is uniformly bounded, thus $\{\alpha_k p_k\}$ converges to zero.

Let $\bar{\mathcal{G}}$ denote the subsequence of $\mathcal{G}$ where the initial step was rejected, i.e., $\bar{\mathcal{G}} = \{k \in \mathcal{G} : j_k > 0\}$. The sequence $\{\alpha_k\}_{k \in \mathcal{G}}$ converges to zero, so $\bar{\mathcal{G}}$ is infinite. For each $k \in \bar{\mathcal{G}}$, let $\sigma_k = \alpha_k / \gamma_C$, i.e., the last rejected trial step length. As this step length fails the Armijo condition,

$$f(x_k + \sigma_k p_k) > f(x_k) + \sigma_k \eta_A \nabla f(x_k)^{\mathrm{T}} p_k \quad \text{for all} \quad k \in \bar{\mathcal{G}}.$$

Equivalently,

$$\begin{aligned} f(x_k + \sigma_k p_k) - f(x_k) - \sigma_k \nabla f(x_k)^{\mathrm{T}} p_k &> -\sigma_k (1 - \eta_A) \nabla f(x_k)^{\mathrm{T}} p_k \\ &> \sigma_k (1 - \eta_A) \varepsilon. \end{aligned} \tag{2.11}$$

Consider the Taylor expansion of $f$ about $x_k + \sigma_k p_k$ using the integral form of the remainder:

$$f(x_k + \sigma_k p_k) - f(x_k) - \sigma_k \nabla f(x_k)^{\mathrm{T}} p_k = \sigma_k \int_0^1 (\nabla f(x_k + t\sigma_k p_k) - \nabla f(x_k))^{\mathrm{T}} p_k dt. \tag{2.12}$$

Let $||\cdot||_\star$ denote the dual norm of $||\cdot||$. Then

$$|(\nabla f(x_k + t\sigma_k p_k) - \nabla f(x_k))^{\mathrm{T}} p_k| \leq ||\nabla f(x_k + t\sigma_k p_k) - \nabla f(x_k)||_\star ||p_k||.$$

Applying this to (2.11) and (2.12) gives

$$(1-\eta_A)\varepsilon < \int_0^1 (\nabla f(x_k + t\sigma_k p_k) - \nabla f(x_k))^{\mathrm{T}} p_k dt \leq \max_{0\leq t\leq 1} ||\nabla f(x_k + t\sigma_k p_k) - \nabla f(x_k)||_\star ||p_k||$$

for each $k \in \bar{\mathcal{G}}$. As $\nabla f$ is continuous, there exists a $t_k^\star$ that achieves the maximum on the right-hand side. Let $\theta_k = t_k^\star \sigma_k = t_k^\star \alpha_k / \gamma_C < \alpha_k$. Then

$$(1-\eta_A)\varepsilon < ||\nabla f(x_k + \theta_k p_k) - \nabla f(x_k)||_\star ||p_k||.$$

Now, $\{\alpha_k p_k\}$ converges to zero, and $\theta_k < \alpha_k$, therefore $\theta_k p_k$ converges to zero as well. The continuity of $\nabla f$ then implies

$$||\nabla f(x_k + \theta_k p_k) - \nabla f(x_k)||_\star \to 0.$$

However, $\varepsilon$ and $\eta_A$ are fixed, so $(1-\eta_A)\varepsilon > 0$. This gives the desired contradiction. □

This result can be refined to take into account how the search direction $p_k$ is chosen. For now, assume that the search direction $p_k$ is chosen to be the steepest-descent direction in the $||\cdot||_2$ norm, i.e., $p_k = -\nabla f(x_k)$. Then the following holds:

**Theorem 2.4.2** (Convergence of steepest-descent with backtracking line-search, [16] Chapter 3)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on an open convex set $\mathcal{D}$. Consider the sequence $\{x_k\}_{k=1}^\infty \subset \mathcal{D}$ generated by the steepest-descent line-search method. If the initial point $x_0 \in \mathcal{D}$ is chosen such that the level set $\mathcal{L}(f(x_0))$ is compact, then either $\nabla f(x_l) = 0$ for some index $l < \infty$, or $\lim_{k\to\infty} \nabla f(x_k) = 0$.*

*Proof.* Let $||\cdot||$ denote the Euclidean 2 norm. By the compactness of $\mathcal{L}(f(x_0))$, the sequence $\{||p_k||\} = \{||\nabla f(x_k)||\}$ is bounded. Thus, by theorem (2.4.1),

$$\lim_{k\to\infty} |\nabla f(x_k)^{\mathrm{T}} p_k| = \lim_{k\to\infty} ||\nabla f(x_k)||_2^2 = 0$$

. □

**Convergence of Newton's methods with backtracking line-search**

It has been shown that when a backtracking line search is used, it holds that $\lim_{k\to\infty} |\nabla f(x_k)^{\mathrm{T}} p_k| = 0$ under mild assumptions of $f$, assuming that $p_k$ is a descent direction at each iteration. Letting $p_k$ be the steepest-descent direction shows that the sequence of gradients also converges to zero. In the case of methods inspired by Newton's method, the situation becomes more complicated. First off, the condition that $\lim_{k\to\infty} |\nabla f(x_k)^{\mathrm{T}} p_k| = 0$ does not guarantee that $\nabla f(x_k) \to 0$. Thus, $p_k$ needs to be chosen in such a manner that $|\nabla f(x_k)^{\mathrm{T}} p_k| \to 0$ only if $\nabla f(x_k) \to 0$. This property of the search direction $p_k$ is independent of the line-search procedure.

**Definition 2.4.1** (Directions of sufficient descent, [16] Chapter 3)**.** A sequence of directions $\{p_k\}$ is a sequence of directions of sufficient descent if $||p_k||$ is bounded and

$$\nabla f(x_k)^{\mathrm{T}} p_k \to 0 \text{ implies } \nabla f(x_k) \to 0 \text{ and } p_k \to 0.$$

The steepest-descent direction $p_k = -\nabla f(x_k)$ clearly satisfies this definition. A more general set of directions of sufficient decrease can be provided by the following lemma.

**Lemma 2.4.3** ([16] Chapter 3)**.** *Let $\{H_k\}$ be a sequence of symmetric positive-definite matrices with bounded condition number, i.e.,*

$$\lambda_{\max}(H_k) \leq M < \infty \text{ and } \lambda_{\min}(H_k) \geq m > 0,$$

*where $M$ and $m$ are constants and $\lambda_{\min}$ and $\lambda_{\max}$ denote the largest and smallest eigenvalues of $H_k$. If $p_k$ is chosen to be the solutions of $H_k p_k = -\nabla f(x_k)$, then $p_k$ is a direction of sufficient descent.*

*Proof.* By the definition of $p_k$,

$$|\nabla f(x_k)^{\mathrm{T}} p_k| = |p_k^{\mathrm{T}} H_k p_k| \geq \lambda_{\min}(H_k)||p_k||^2 \geq m||p_k||^2.$$

Thus, if $|\nabla f(x_k)^{\mathrm{T}} p_k| \to 0$, $p_k \to 0$. Furthermore,

$$||H_k||\ ||p_k|| \geq ||\nabla f(x_k)||,$$

or equivalently,

$$||p_k|| \geq \frac{||\nabla f(x_k)||}{||H_k||}.$$

Now, $||H_k|| = \lambda_{\max}(H_k) \leq M$, thus $p_k \to 0$ implies that $\nabla f(x_k) \to 0$. □

This notion of directions of sufficient decrease is enough to show the global convergence of Newton's method with a backtracking line search.

**Theorem 2.4.4** (Global Convergence of Newton-based method, [16] Chapter 3)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable on an open convex set $\mathcal{D} \subseteq \mathbb{R}^n$. Assume that $f$ is strongly convex, i.e., that there exists a uniform lower bound on the lowest eigenvalue of $\nabla^2 f(x)$ for all $x \in \mathcal{D}$, i.e.,*

$$d^{\mathrm{T}} \nabla^2 f(x) d \geq m||d||_2^2 \text{ for some } \quad m > 0.$$

*Given $x_0$ such that $\mathcal{L}(f(x_0))$ is compact, consider the sequence of iterates $x_{k+1} = x_k + \alpha_k p_k$, where $p_k$ satisfies $\nabla^2 f(x_k) p_k = -\nabla f(x_k)$ and $\alpha_k$ satisfies the Armijo condition. Then the sequence $\{x_k\}$ is well-defined and lies in $\mathcal{L}(f(x_0))$, and the algorithm finds some $x_k$ that meets a convergence criterion, or* $\lim_{k\to\infty} \nabla f(x_k) = 0$.

*Proof.* Given $x_k \in \mathcal{L}(f(x_0))$, the positive definiteness of $\nabla^2 f(x_k)$ ensures that $p_k$ is a descent direction at every iterate. Thus, there exists an $\alpha_k$ that satisfies the Armijo condition, so $x_{k+1} \in \mathcal{L}(f(x_0))$. By assumption, the lowest eigenvalue of $\nabla^2 f(x_k)$ is bounded away from zero. By the continuity of $\nabla^2 f(x)$ and the compactness of $\mathcal{L}(f(x_0))$, $||\nabla^2 f(x_k)||_2$ is bounded, i.e., the largest eigenvalue is bounded above by some constant $M$. Thus, $p_k$ is a direction of sufficient decrease, so the method is convergent. □

The next theorem shows that the initial choice of step length $\alpha_k = 1$ eventually will produce a sufficient decrease in $f$.

**Theorem 2.4.5** (Sufficient decrease with unit step, [16] Chapter 3)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable on an open convex set $\mathcal{D} \subseteq \mathbb{R}^n$. Assume that $f$ is strongly convex, i.e., that there exists a uniform lower bound on the lowest eigenvalue of $\nabla^2 f(x)$ for all $x \in \mathcal{D}$, i.e.,*

$$d^{\mathrm{T}} \nabla^2 f(x) d \geq m||d||_2^2 \text{ for some } \quad m > 0.$$

*Given $x_0$ such that $\mathcal{L}(f(x_0))$ is compact, consider the sequence of iterates $x_{k+1} = x_k + \alpha_k p_k$, where $p_k$ satisfies $\nabla^2 f(x_k) p_k = -\nabla f(x_k)$ and $\alpha_k$ satisfies the Armijo condition with the Armijo parameter $\eta_A < \frac{1}{2}$. Then there exists an index $K$ such that for all $k \geq K$ for which convergence has not been achieved*

*($\nabla f(x_k) \neq 0$), the step $\alpha_k = 1$ satisfies the Armijo condition*

$$f(x_k) - f(x_k + \alpha p_k) \geq -\eta_a \alpha \nabla f(x_k)^{\mathrm{T}} p_k.$$

*Proof.* Notice that

$$-\nabla f(x_k)^{\mathrm{T}} p_k = p_k^{\mathrm{T}} \nabla^2 f(x_k) p_k \geq m||p_k||^2.$$

It has been shown that $\nabla f(x_k) \to 0$ implies that the sequence of Newton direction $\{p_k\}$ converges to zero. This condition, along with the descent property of $p_k$, implies that for $k$ sufficiently large, $x_k + p_k \in \mathcal{L}(f(x_0))$. Assume that index $k$ is such an index where this is true. It remains to show that the Armijo condition is satisfied with $\alpha_k = 1$.

Consider the Taylor expansion of $f$ about $x$ with the integral form of the remainder

$$f(x+p) = f(x) + \nabla f(x)^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2 f(x) p + \int_0^1 p^{\mathrm{T}} \left( \nabla^2 f(x + tp) - \nabla^2 f(x) \right) p (1-t) dt.$$

Substituting the definition of $p_k$ gives

$$\begin{aligned} f(x_k + p_k) - f(x_k) - \eta_A \nabla f(x)^{\mathrm{T}} p_k = \frac{1}{2}(1 - 2\eta_A) \nabla f(x_k)^{\mathrm{T}} p_k \\ + \int_0^1 p_k^{\mathrm{T}} \left( \nabla^2 f(x + tp) - \nabla^2 f(x) \right) p_k (1-t) dt. \end{aligned}$$

Let

$$\omega_k = \max_{0 \leq t \leq 1} ||\nabla^2 f(x_k + t p_k) - \nabla^2 f(x_k)||.$$

Due to the fact that $\mathcal{L}(f(x_0))$ is compact and $\nabla^2 f(x)$ is continuous, $\omega_k$ is bounded. Then

$$\begin{aligned} f(x_k + p_k) - f(x_k) - \eta_A \nabla f(x_k)^{\mathrm{T}} p_k &\leq \frac{1}{2}(1 - 2\eta_A) \nabla f(x_k)^{\mathrm{T}} p_k + \frac{1}{2} \omega_k ||p_k||^2 \\ &\leq \frac{1}{2}(1 - 2\eta_A - \frac{\omega_k}{m}) \nabla f(x_k)^{\mathrm{T}} p_k. \end{aligned}$$

As $p_k \to 0$, there must exist an index $K$ such that for all $k \geq K$, $\omega_k$ will be sufficiently small so that $\omega_k < m(1 - 2\eta_A)$. Assuming that $\eta_A < 1/2$, then the term on the right-hand side will be negative. Thus, the Armijo condition is satisfied with $\alpha_k = 1$ for $k \geq K$. □

The following theorem demonstrates that, under certain conditions, Newton's method with unit step size achieves quadratic convergence.

**Theorem 2.4.6** (Quadratic convergence of Netwon's method, [16] Chapter 3)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$*

*is twice continuously differentiable and that the Hessian matrix $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of a solution $x^*$ at which the second order sufficient conditions are satisfied. Consider the Newton iterations $x_{k+1} = x_k + p_k$, where $p_k$ satisfies $\nabla^2 f(x_k)p_k = -\nabla f(x_k)$. Then,*

1. *If the starting point $x_0$ is sufficiently close to $x^*$, $x_k \to x^*$,*
2. *The rate of convergence is quadratic, and*
3. *The sequence $\{||\nabla f(x_k)||\}$ converges to zero quadratically.*

*Proof.* Recall the optimality condition $\nabla f(x^*) = 0$. Consider the error in $x_{k+1}$ and the optimal solution $x^*$:

$$\begin{aligned} x_k + p_k - x^* &= x_k - x^* - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \\ &= \nabla^2 f(x_k)^{-1} \left( \nabla^2 f(x_k)(x_k - x^*) - (\nabla f(x_k) - \nabla f(x^*)) \right) \end{aligned}$$

By Taylor's theorem, it holds that

$$\nabla f(x_k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x_k + t(x^* - x_k))(x_k - x^*) dt.$$

Putting these two equations together yields

$$\begin{aligned} &||\nabla^2 f(x_k)(x_k - x^*) - (\nabla f(x_k) - \nabla f(x^*))|| \\ &\qquad = || \int_0^1 \left( \nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k)) \right) (x_k - x^*) dt|| \\ &\qquad \leq \int_0^2 ||\nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k))||\ ||x_k - x^*|| dt \\ &\qquad \leq ||x_k - x^*||^2 \int_0^1 L t dt = \frac{1}{2} L ||x_k - x^*||^2, \end{aligned}$$

where $L$ is the local Lipschitz constant of $\nabla^2 f(x)$. As $\nabla^2 f(x^*)$ is nonsingular, and $\nabla^2 f(x)$ is continuous, there exists a radius $r$ such that $||\nabla^2 f(x_k)^{-1}|| \leq 2||\nabla^2 f(x^*)^{-1}||$ for all $x_k$ such that $||x_k - x^*|| \leq r$. Therefore,

$$||x_k + p_k - x^*||_2 \leq L||\nabla^2 f(x^*)^{-1}||\ ||x_k - x^*||^2.$$

If $x_0$ is chosen so that $||x_0 - x^*|| \leq \min(r, 1/(2L||\nabla^2 f(x^*)^{-1}||))$, then it can inductively be shown that the sequence converges quadratically to $x^*$.

Now, consider the relations $x_{k+1} - x_k = p_k$ and $\nabla f(x_k) + \nabla^2 f(x_k) p_k = 0$. Then

$$\begin{aligned}
||\nabla f(x_{k+1})|| &= ||\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k) p_k|| \\
&= || \int_0^1 \nabla^2 f(x_k + t p_k)(x_{k+1} - x_k) dt - \nabla^2 f(x_k) p_k|| \\
&\leq \int_0^1 ||\nabla^2 f(x_k + t p_k) - \nabla^2 f(x_k)||\, ||p_k|| dy \\
&\leq \frac{1}{2} L ||p_k||^2 \\
&\leq \frac{1}{2} L ||\nabla^2 f(x_k)^{-1}||^2 ||\nabla f(x_k)||^2 \\
&\leq 2L ||\nabla^2 f(x^*)^{-1}||^2 ||\nabla f(x_k)||^2,
\end{aligned}$$

proving that the norm of the gradients converges quadratically. □

Note that this proof only shows convergence to stationary points, not necessarily to minimizers. The following theorem shows that if $f$ satisfies additional properties, Newton's method converges quadratically to a local minimizer.

**Theorem 2.4.7** ([5]). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable. Assume that the Hessian matrix $\nabla^2 f(x)$ is Lipschitz continuous with Lipschitz constant $L$ and that the eigenvalues of $\nabla^2 f(x)$ are bounded above and below, i.e., there exist constants $m$ and $M$ such that*

$$mI \preceq \nabla f(x) \preceq MI.$$

*Then Newton's method with a backtracking line search is globally convergent. Furthermore, an index $K$ exists such that, for all $k \geq K$, the backtracking line search accepts $\alpha = 1$, and the method converges quadratically once $k \geq K$.*

*Proof.* Let $\eta$ be some number such that $0 < \eta \leq m^2/L$. Assume that, at iteration $k$, $||\nabla f(x_k)||_2 \geq \eta$. First, a lower bound on the step size selected by the backtracking line search is derived when this is the case. By assumption, $\nabla^2 f(x_k) \prec MI$, and therefore,

$$\begin{aligned}
f(x_k + \alpha_k p_k) &\leq f(x_k) + \alpha_k \nabla f(x_k)^{\mathrm{T}} p_k + \frac{M}{2} \alpha_k^2 ||p_k||_2^2 \\
&\leq f(x) - \alpha_k \lambda(x_k)^2 + \frac{M}{2m} \alpha_k^2 \lambda(x_k)^2,
\end{aligned}$$

where $\lambda(x)^2 = p_k^{\mathrm{T}} \nabla^2 f(x_k) p_k = (\nabla f(x_k))^{\mathrm{T}} (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$ is the *Newton decrement*. Consider the

step size $\widehat{\alpha} = m/M$. Then $\widehat{\alpha}$ satisfies the Armijo condition, as

$$f(x_k + \widehat{\alpha} p_k) \leq f(x_k) - \frac{m}{2M}\lambda^2(x_k) \leq f(x) - \eta_A \widehat{\alpha}\lambda(x)^2.$$

Therefore, the backtracking line search yields a step $\alpha_k \geq \gamma_C m/M$. The reduction in the objective is then

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\leq -\eta_A \alpha_k \lambda(x_k)^2 \\ &\leq -\eta_A \gamma_C \frac{m}{M}\lambda(x_k)^2 \\ &\leq -\eta_A \gamma_C \frac{m}{M^2}||\nabla f(x_k)||_2^2 \\ &\leq -\eta_A \gamma_C \eta^2 \frac{m}{M^2}. \end{aligned}$$

Let $\gamma = \eta_A \gamma_C \eta^2 \frac{m}{M^2}$. Then

$$f(x_{k+1}) - f(x_k) \leq -\gamma \tag{2.13}$$

when $||\nabla f(x_k)||_2 \geq \eta$. Thus, if the gradient at $x_k$ is sufficiently large, the backtracking line search will produce a reduction in the objective at least as large as $\gamma$.

Now, suppose that $||\nabla f(x_k)||_2 < \eta$. Furthermore, assume that

$$\eta \leq 3(1 - 2\eta_A)\frac{m^2}{L}.$$

By the Lispchitz continuity of the second derivative, it holds for all positive step-sizes $\alpha$ that

$$||\nabla^2 f(x_k + \alpha p_k) - \nabla^2 f(x_k)||_2 \leq \alpha L||p_k||_2.$$

Therefore, it holds that

$$|p_k^{\mathrm{T}}(\nabla^2 f(x_k + \alpha p_k) - \nabla^2 f(x_k))p_k| \leq \alpha L||p_k||_2^3. \tag{2.14}$$

Let $\tilde{f}_k(\alpha) = f(x_k + \alpha p_k)$, so that $\tilde{f}_k''(\alpha) = p^{\mathrm{T}}\nabla^2 f(x_k + \alpha p_k)p_k$. Thus, (2.14) can be written as

$$|\tilde{f}_k''(\alpha) - \tilde{f}_k''(0)| \leq \alpha L||p_k||_2^3. \tag{2.15}$$

After some rearrangement, this becomes

$$\tilde{f}_k''(\alpha) \leq \tilde{f}_k''(0) + \alpha L||p_k||_2^3 \leq \lambda(x_k)^2 + \alpha\frac{L}{m^{3/2}}\lambda(x_k)^3.$$

Integrating the above inequality yields

$$\begin{aligned}\tilde{f}_k'(\alpha) &\leq \tilde{f}_k'(0) + \alpha\lambda(x_k)^2 + \alpha^2\frac{L}{2m^{3/2}}\lambda(x_k)^3 \\ &\leq -\lambda(x_k)^2 + \alpha\lambda(x_k)^2 + \alpha^2\frac{L}{2m^{3/2}}\lambda(x_k)^3,\end{aligned}$$

where the second inequality uses the fact that $\tilde{f}_k'(0) = -\lambda(x_k)^2$. Integrating again yields

$$\tilde{f}_k(\alpha) \leq \tilde{f}_k(0) - \alpha\lambda(x_k)^2 + \frac{\alpha^2}{2}\lambda(x_k)^2 + \alpha^3\frac{L}{6m^{3/2}}\lambda(x_k)^3.$$

Taking $\alpha = 1$ gives

$$f(x_k + p_k) \leq f(x_k) - \frac{1}{2}\lambda(x_k)^2 + \frac{L}{6m^{3/2}}\lambda(x_k)^3. \tag{2.16}$$

Consider the assumption that $||\nabla f(x_k)||_2 \leq \eta \leq 3(1-2\eta_A)m^2/L$. By strong convexity and the definition of the Newton decrement, it holds that

$$\lambda(x_k) \leq 3(1-2\eta_A)m^{3/2}/L.$$

Combining this with (2.16) yields

$$\begin{aligned}f(x_k + p_k) &\leq f(x_k) - \frac{1}{2}\lambda(x_k)^2\left(\frac{1}{2} - \frac{L\lambda(x_k)}{6m^{3/2}}\right) \\ &\leq f(x_k) - \eta_A\lambda(x_k)^2 \\ &= f(x_k) + \eta_A\nabla f(x_k)^{\mathrm{T}}p_k.\end{aligned}$$

Thus, the backtracking line search accepts the unit step. Next, the rate of convergence is analyzed. Consider the following:

$$\begin{aligned}||f(x_k + p_k)||_2 &= ||f(x_k + p_k) - \nabla f(x_k) - \nabla^2 f(x_k)p_k||_2 \\ &= \left|\left|\int_0^1 (\nabla^2 f(x_k + tp_k) - \nabla^2 f(x))p_k dt\right|\right|_2 \\ &\leq \frac{L}{2}||p_k||_2^2 \\ &= \frac{L}{2}||\nabla^2 f(x_k)^{-1}\nabla f(x_k)||_2^2 \\ &\leq \frac{L}{2m^2}||\nabla f(x)||_2^2\end{aligned} \tag{2.17}$$

Note that (2.17) implies that if $||\nabla f(x_k)||_2 \leq \eta \leq m^2/L$, then $||f(x_k + p_k)||_2 \leq \eta$ as well. It follows then

that the method will select the full Newton step for the $k$th and all subsequent iterations and converge quadratically if $||\nabla f(x_k)|| < \eta = \min\{1, 3(1-2\eta_A)\}m^2/L$. If this does not hold, the objective function is still guaranteed to decrease by at least $\gamma$ each time the Newton step is not selected. Thus, there can only be finitely many steps at which $||\nabla f(x_k)||_2 \geq \eta$. The result follows.

□

Unfortunately, it is not always the case that the objective function is strongly convex. For certain functions, there is no guarantee that the Hessian matrices of $f$ are positive definite. In this case, the Newton direction may not even be a descent direction. This opens up a wide array of potential algorithms that fall under the category of *modified Newton methods*, in which the search directions $p_k$ are determined via $(\nabla^2 f(x_k) + E_k)p_k = \nabla f(x_k)$, where $E_k$ is some symmetric positive definite matrix such that $(\nabla^2 f(x_k) + E_k)$ is sufficiently positive definite. These methods can be shown to still be globally convergent, and if $E_k$ is chosen to be zero if $\nabla^2 f(x_k)$ is already sufficiently positive definite, then quadratic convergence can still be achieved. There exists an enormous amount of literature on modified Newton's methods. In the next section, an alternative family of methods is analyzed: trust-region methods. Trust-region methods have the benefit of being able to work with the Hessian of $f$ without needing to perform some sort of modification. Instead, convergence is guaranteed by minimizing a sequence of potentially non-convex quadratic models subject to a convex constraint.

## 2.5 The Trust-Region Method

Trust-region methods provide an alternative method of utilizing the local quadratic model of a function $f$ without modifications or assuming that the quadratic model is positive definite. In certain situations, the distinction between line-search methods and trust-region methods becomes quite blurry, and as the understanding of the two methods has grown, the distinction between the two has shrunk.

A basic trust-region algorithm is thoroughly analyzed in this section. This method already shares some similarities with backtracking line searches. As before, a sequence of iterates $\{x_k\}$ is computed with an update rule of the form $x_{k+1} = x_k + p_k$, where a sufficient decrease condition is enforced by comparing the actual reduction $f(x_k) - f(x_{k+1})$ with the predicted reduction $m_k(x_k) - m_k(x_k + p_k)$, where $m_k$ is a local model of $f$. As in a line-search method, should the sufficient decrease condition fail, the norm of the search direction $||p_k||$ is reduced. However, in the trust-region case, instead of merely contracting the step by reducing the step length, the direction of the vector $p_k$ may be changed as well. This is done by explicitly limiting the norm of the vector $p_k$ and by defining it as a minimizer of the constrained quadratic

problem

$$\min_{p \in \mathbb{R}^n} q_k(x_k + p) \quad \text{subject to} \quad ||p|| \leq \delta_k,$$

where $q_k$ is the local quadratic model of $f$ about $x_k$, given by

$$q_k(x) = f(x_k) + g_k(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}} H_k (x - x_k), \tag{2.18}$$

with $g_k = \nabla f(x_k)$ and $H_k \approx \nabla^2 f(x_k)$. Note that the norm $||\cdot||$ in the constraint is arbitrary. The scalar quantity $\delta_k$ is referred to as the *trust-region radius*, and the set $\{p : ||p|| \leq \delta_k\}$ is referred to as the *trust region*. Once the step $p_k$ is computed, the actual versus predicted reduction ratio is computed as

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}.$$

A typical trust-region method chooses the model $m_k$ in the definition of $\rho_k$ to be $q_k$, however, other choices are possible. To simplify the notation, let $Q_k(p) = q_k(x_k + p) - q_k(x_k)$. This test deviates from the Armijo condition, which uses a linear model function in the definition of $\rho_k$. If $\rho_k \geq \eta_A$, then the trust-region step is accepted, setting $x_{k+1} = x_k + p_k$. Should the test fail, i.e., $\rho_k < \eta_A$, then the trust-region radius is reduced by some constant factor $\gamma_C$, and the trust-region subproblem is solved again with the new radius. This method already bears some similarities to backtracking line searches, except now there is the additional computational work of approximately solving the trust-region subproblem.

In order for the method to be competitive with the backtracking line-search method, the trust region needs to have the opportunity to expand as well as shrink, otherwise, the method will fall into the trap of simply taking arbitrarily small steps at each iteration. To achieve this goal, a second sufficient decrease parameter $\eta_E$ is chosen such that $\eta_A < \eta_E < 1$. If $\rho_k \geq \eta_E$, then the local quadratic model is assumed to be a very good model of the objective function $f$, so the trust-region radius is expanded by some constant factor $\gamma_E$. This motivates the basic trust-region algorithm, presented in Algorithm 2.2.

Up until this point, the choice of norm used to define the trust region has not been specified. Two of the most natural choices are the polygonal norms $||\cdot||_1$ and $||\cdot||_\infty$, and elliptic norms (including the Euclidean norm). Unfortunately, polygonal norm trust-region subproblems have a number of theoretical and practical difficulties associated with them, as they are instances of linearly-constrained quadratic problems. In what follows, practical implementations of Algorithm 2.2 are restricted to use elliptic norms, i.e., $||p|| = ||p||_B = (p^{\mathrm{T}} B p)^{1/2}$ for some positive definite matrix $B$. In the unconstrained optimization setting, $B$ is typically chosen as $I$, however when the trust-region method is applied to a constrained

**Algorithm 2.2.** Basic Trust-Region Algorithm

Given constants $\eta_A$, $\eta_E$, $\gamma_C$, $\gamma_E$, $\delta_0$ such that $0 < \eta_A < \eta_E < 1$, $\eta_A < 1/2$, $0 < \gamma_C < 1 < \gamma_E$, and $\delta_0 > 0$
$k \leftarrow 0$
**while** Not converged **do**
    $p_k \approx \text{argmin}_{p \in \mathbb{R}^n} \{q_k(x_k + p) \ : \ ||p||_k \leq \delta_k\}$
    $\rho_k = (f(x_k) - f(x_k + p_k))/(q_k(x_k) - q_k(x_k + p_k))$
    **if** $\rho_k \geq \eta_A$ **then**
        $x_{k+1} = x_k + p_k$
        **if** $\rho_k \geq \eta_E$ **then**
            $\delta_{k+1} \leftarrow \max\{\delta_k, \gamma_E ||p_k||\}$
        **else**
            $\delta_{k+1} \leftarrow \delta_k$
        **end if**
    **else**
        $x_{k+1} \leftarrow x_k$
        $\delta_{k+1} \leftarrow \gamma_C ||p_k||$
    **end if**
    $k \leftarrow k + 1$
**end while**

problem, the matrix $B$ is varied at each iteration. In the analysis, it shall be assumed that the norm $||\cdot|| = ||\cdot||_k$ is changing at each iteration and that the family of norms $\{||\cdot||_k\}$ satisfies certain conditions that ensure the algorithm converges.

**The steepest-descent approximation to the trust-region subproblem**

When using the basic trust-region method 2.2 to solve an optimization problem, it is crucial that not too much effort be taken to solve the trust-region subproblem at each iteration. At the same time, whatever method is used to compute an approximate solution to the subproblem must yield a solution accurate enough that it does not interfere with the progress of the overall algorithm. Steepest-descent directions play a vital role in defining viable approximations to the trust-region subproblem. It will be shown that the convergence of the trust-region method will be guaranteed as long as the approximate solution gives a predicted reduction in the objective function that is at least a fixed fraction of the reduction predicted by the steepest-descent step scaled to lie within the trust-region.

**Definition 2.5.1** (Gradient-descent Cauchy point, [7] Chapter 6, [16] Chapter 3)**.** Let $||\cdot||$ be *any* norm. Consider the trust-region problem.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & Q(p) = \frac{1}{2} p^{\mathrm{T}} H x + g^{\mathrm{T}} p \\ \text{subject to} \quad & ||p|| \leq \delta. \end{aligned} \tag{2.19}$$

Then the Cauchy point of the above problem is defined as

$$\underset{p\in\mathbb{R}^n,\alpha\in\mathbb{R}}{\operatorname{argmin}} \; Q(p) = \frac{1}{2}p^{\mathrm{T}}Hp + g^{\mathrm{T}}p$$
$$\text{subject to} \quad ||p|| \leq \delta \quad \text{and} \quad p = -\alpha g.$$

Let $p^C$ denote the Cauchy point as in Definition 2.5.1, and let $\alpha^C$ denote the step length used to scale the Cauchy point, i.e., the scalar so that $p^C = -\alpha^C g$. Then

$$\alpha^C = \begin{cases} \dfrac{g^{\mathrm{T}}g}{g^{\mathrm{T}}Hg} & \text{if } \dfrac{g^{\mathrm{T}}g}{g^{\mathrm{T}}Hg} \leq \dfrac{\delta}{||g||_B} \text{ and } g^{\mathrm{T}}Hg > 0; \\ \dfrac{\delta}{||g||_B} & \text{otherwise.} \end{cases}$$

This definition can be used to create an upper bound for the objective function $Q(p^C)$.

**Lemma 2.5.1** ([16] Chapter 3)**.** *Let $p^C$ be the Cauchy point along the search direction $-g$ of problem* (2.19)*. Then,*

$$Q(p^C) \leq -\frac{1}{2}||g||_2 \min\{\delta\frac{||g||_2}{||g||}, \frac{||g||_2}{||H||_2}\}$$

*Proof.* First, consider the case where $g^{\mathrm{T}}Hg > 0$. Then $Q(-\alpha g) = \frac{1}{2}\alpha^2 g^{\mathrm{T}}Hg - \alpha g^{\mathrm{T}}g$. This has a unique unconstrained minimizer at $\alpha^C = \alpha^\star = g^{\mathrm{T}}g/g^{\mathrm{T}}Hg$. Then

$$Q(-\alpha^C g) = -\frac{1}{2}\frac{(g^{\mathrm{T}}g)^2}{g^{\mathrm{T}}Hg} \leq -\frac{1}{2}\frac{||g||_2^2}{||H||_2}.$$

Next, consider the case where $g^{\mathrm{T}}g/g^{\mathrm{T}}Hg > \delta/||g||$, so that $\alpha^C = \delta/||g|| \leq \alpha^\star$, and

$$\begin{aligned} Q(-\alpha^C g) =& -\alpha^C g^{\mathrm{T}}g + \frac{1}{2}(\alpha^C)^2 g^{\mathrm{T}}Hg \\ \leq& -\alpha^C g^{\mathrm{T}}g + \frac{1}{2}\alpha^C\alpha^\star g^{\mathrm{T}}Hg \\ =& -\frac{1}{2}g^{\mathrm{T}}g\left(\frac{\delta}{||g||}\right) = -\frac{1}{2}||g||_2\left(\delta\frac{||g||_2}{||g||}\right). \end{aligned}$$

Finally, consider the case where $g^{\mathrm{T}}Hg \leq 0$, so no minimizer exists along $-g$, and $\alpha^C = \delta/||g||$. Then

$$Q(-\alpha^C g) = -\alpha^C g^{\mathrm{T}}g + \frac{1}{2}(\alpha^C)^2 g^{\mathrm{T}}Hg \leq -\alpha^C g^{\mathrm{T}}g \leq -\frac{1}{2}||g||_2\left(\delta\frac{||g||_2}{||g||}\right).$$

The result follows. □

**Corollary 2.5.1.1.** *Let $||\cdot||$ denote any norm, and let $c > 0$ be a constant such that $||d||_2 \geq c||d||$. Then*

$$Q(p^C) \leq -\frac{1}{2}c^2||g|| \min\Big\{\delta, \frac{||g||}{||H||_2}\Big\}.$$

Defining the Cauchy step as in Definition 2.5.1 is sufficient to prove the theoretical convergence of Algorithm 2.2. However, it is fairly unintuitive to define the Cauchy point of a trust-region problem in which an arbitrary norm defines the shape of the trust region in terms of the gradient-descent direction. Instead, the Cauchy step can be defined using the steepest-descent direction induced by the norm which defines the trust region. Recall that, given a norm $||\cdot||$, it's dual norm $||\cdot||_\star$, and a vector $g = \nabla f(x)$ at some point $x$, the steepest-descent direction at $x$ is defined as

$$h = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\{g^{\mathrm{T}}x : ||x|| = ||g||_\star\}, \tag{2.20}$$

with $h = -g$ when the norm is the 2-norm. Note that the Cauchy-Schwartz inequality implies $h$ is the vector that satisfies $-||g||_\star||h|| = g^{\mathrm{T}}h$. Combining this with (2.20) gives $||g||_\star^2 = -g^{\mathrm{T}}h$. This leads to the following more general definition of the Cauchy point $p^C$.

**Definition 2.5.2** (Cauchy Point 2)**.** Let $||\cdot||$ be any norm, with dual norm $||\cdot||_\star$. The Cauchy point of problem (2.19) along the steepest-descent direction $h$ induced by $g$ and $||\cdot||$ is defined as

$$\begin{aligned} &\underset{x\in\mathbb{R}^n,\alpha\in_R}{\operatorname{argmin}} \; Q(p) = \frac{1}{2}p^{\mathrm{T}}Hp + g^{\mathrm{T}}p \\ &\text{subject to} \quad ||p|| \leq \delta \quad \text{and} \quad p = \alpha h. \end{aligned}$$

This definition is more often used when defining an approximate solution to be used in a specific implementation, whereas the previous definition more commonly appears in proofs of convergence. The step length can similarly be defined as

$$\alpha^C = \begin{cases} \dfrac{||g||_\star^2}{h^{\mathrm{T}}Hh} & \dfrac{||g||_\star^2}{h^{\mathrm{T}}Hh} \leq \dfrac{\delta}{||g||_\star} \;\text{ and }\; h^{\mathrm{T}}Hh > 0 \\ \dfrac{\delta}{||g||_\star} & \text{otherwise} \end{cases},$$

**Lemma 2.5.2.** *Let $p^C$ be the Cauchy point along the direction the steepest-descent direction $h$ induced by $g$ and $||\cdot||$. Then,*

$$Q(p^C) \leq -\frac{1}{2}||g||_\star \min\{\delta, ||g||_\star/||H||\},$$

*where* $||H|| = \max_{x \neq 0} \dfrac{||Hx||_\star}{||x||}$.

The proof is essentially identical to the proof of lemma (2.5.1). Finally, a third definition is presented in which the search direction is *any* steepest-descent direction. This definition is a generalization of Definitions 2.5.1 and 2.5.2.

**Definition 2.5.3** (Cauchy Point Definition 3)**.** Let $||\cdot||$ and $||\cdot||'$ be any two norms, with dual norms $||\cdot||_\star$ and $||\cdot||'_\star$, respectively. Let $h$ denote the steepest-descent direction induced by $||\cdot||'$ and the vector $g$. Then the Cauchy step along $h$ is given by

$$\begin{aligned} &\underset{x \in \mathbb{R}^n, \alpha \in R}{\operatorname{argmin}} \; Q(p) = \frac{1}{2} p^{\mathrm{T}} H p + g^{\mathrm{T}} p \\ &\text{subject to} \quad ||p|| \leq \delta \quad \text{and} \quad p = \alpha h. \end{aligned}$$

**Lemma 2.5.3.** *Let $p^C$ be the Cauchy point along the direction the steepest-descent direction $h$ with respect to some norm $||\cdot||'$ not necessarily identical to the trust-region norm $||\cdot||$. Then,*

$$Q(p^C) \leq -\frac{1}{2} ||g||'_\star \min\{\delta \frac{||g||'_\star}{||h||}, \frac{||g||'_\star}{||H||'}\},$$

*where* $||H||' = \max_{x \neq 0} \frac{||Hx||'_\star}{||x||'}$.

*Proof.* To begin, recall that $h$ satisfies $-g^{\mathrm{T}} h = ||h||' ||g||'_\star = (||g||'_\star)^2$. Consider the case where $h^{\mathrm{T}} H h > 0$. Then $Q(\alpha h) = \frac{1}{2}\alpha^2 h^{\mathrm{T}} H h + \alpha g^{\mathrm{T}} h$. Let $\alpha^\star$ denote the unique unconstrained minimizer along this arc. Then $\alpha^\star = -g^{\mathrm{T}} h / h^{\mathrm{T}} H h = (||g||'_\star)^2 / h^{\mathrm{T}} H h$, and

$$\begin{aligned} Q(\alpha^\star h) &= \frac{1}{2} \left( \frac{(||g||'_\star)^2}{h^{\mathrm{T}} H h} \right)^2 h^{\mathrm{T}} H h - \left( \frac{(||g||'_\star)^2}{h^{\mathrm{T}} H h} \right) (||g||'_\star)^2 \\ &= -\frac{1}{2} \frac{(||g||'_\star)^4}{h^{\mathrm{T}} H h} \leq -\frac{1}{2} \frac{(||g||'_\star)^2}{||H||'}. \end{aligned}$$

If $\alpha^C = \alpha^\star$, then the result holds. Assume then that this is not the case so that $\alpha^C = \delta / ||h|| \leq \alpha^\star$. Then

$$\begin{aligned} Q(\alpha^C h) &= \alpha^C g^{\mathrm{T}} h + \frac{1}{2} (\alpha^C)^2 h^{\mathrm{T}} H h \\ &\leq -\alpha^C (||g||'_\star)^2 + \frac{1}{2} \alpha^C \alpha^\star h^{\mathrm{T}} H h \\ &= -\frac{\delta}{||h||} (||g||'_\star)^2 + \frac{1}{2} \frac{\delta}{||h||} \frac{(||g||'_\star)^2}{h^{\mathrm{T}} H h} h^{\mathrm{T}} H h \\ &= -\frac{1}{2} ||g||'_\star \delta \frac{||g||'_\star}{||h||}. \end{aligned}$$

Finally, consider the case where $h^{\mathrm{T}}Hh \leq 0$. Then $\alpha^C = \delta/||h||$ and

$$Q(\alpha^C h) = \alpha^C g^{\mathrm{T}}h + \frac{1}{2}(\alpha^C)^2 h^{\mathrm{T}}Hh \leq -\alpha^C(||g||'_\star)^2 \leq -\frac{1}{2}||g||'_\star \delta \frac{||g||'_\star}{||h||}.$$

The result follows from combining all three cases. □

This new third definition reflects the case that arises in the proposed methods for solving the trust-region subproblem for large-scale problems. These methods can trivially be shown to select a search direction that achieves a reduction in the model function at least as large as the reduction predicted by the steepest-descent direction induced by norms that are identical to neither the 2-norm nor the trust-region norm.

Before analyzing Algorithm 2.2 using Definition 2.5.3, it is crucial to understand the concept of a *uniformly equivalent* family of norms, as well as establish some results about such families. To begin, a straightforward result about how the equivalence of two norms relates to the equivalence of their duals is presented.

**Lemma 2.5.4.** *Let $||\cdot||$ and $||\cdot||'$ be two norms norm such that $c_1||x|| \leq ||x||' \leq c_2||x||$ for all $x \in \mathbb{R}^n$, with corresponding dual norm $||\cdot||_\star$ and $||\cdot||'_\star$. Then*

$$\frac{1}{c_2}||y||_\star \leq ||y||'_\star \leq \frac{1}{c_1}||y||_\star$$

*for all $y \in \mathbb{R}^n$.*

*Proof.* Let $y \in \mathbb{R}^n$. Then

$$||y||_\star = \max_{x \neq 0} \frac{|y^{\mathrm{T}}x|}{||x||} \leq \max_{x \neq 0} \frac{||y||'_\star ||x||'}{||x||} \leq c_2||y||'_\star.$$

Similarly,

$$||y||'_\star = \max_{x \neq 0} \frac{|y^{\mathrm{T}}x|}{||x||'} \leq \max_{x \neq 0} \frac{||y||_\star ||x||}{||x||'} \leq \frac{1}{c_1}||y||_\star.$$

The result follows. □

**Definition 2.5.4** (Uniform equivalence of norms)**.** Let $\{||\cdot||_k\}_{k \in \mathcal{K}}$ be a family of norms indexed by some potentially infinite set $\mathcal{K}$. Let $||\cdot||$ denote some other norm. The family of norms $\{||\cdot||_k\}_{k \in \mathcal{K}}$ is *uniformly*

*equivalent* to $||\cdot||$ if there exists positive constants $c_1$ and $c_2$ such that

$$c_1||x||_k \leq ||x|| \leq c_2||x||_k$$

for all $x \in \mathbb{R}^n$ and $k \in \mathcal{K}$.

**Lemma 2.5.5.** *Suppose that $\{||\cdot||_k\}_{k\in\mathcal{K}}$ if uniformly equivalent to $||\cdot||$ with constants $c_1$ and $c_2$. Then the family of dual norms $\{||\cdot||_{k\star}\}$ is uniformly equivalent to $||\cdot||_\star$, i.e.,*

$$\frac{1}{c_2}||y||_{k\star} \leq ||y||_\star \leq \frac{1}{c_1}||y||_{k\star}.$$

.

*Proof.* The result follows from Lemma 2.5.4 and the definition of uniform equivalence. □

**Lemma 2.5.6.** *Let $\{||\cdot||\}_{k\in\mathcal{K}}$ be a family of norms uniformly equivalent to some norm $||\cdot||$. Then $\{||\cdot||\}_{k\in\mathcal{K}}$ is uniformly equivalent to $||\cdot||_j$ for all $j \in \mathcal{K}$.*

*Proof.* Let $j \in \mathcal{K}$, and $x \in \mathbb{R}^n$. Then

$$\frac{1}{c_2}||x|| \leq ||x||_j \leq \frac{1}{c_1}||x||.$$

Furthermore, it holds that

$$\frac{c_1}{c_2}||x||_k \leq \frac{1}{c_2}||x|| \leq ||x||_j \leq \frac{1}{c_1}||x|| \leq \frac{c_2}{c_1}||x||_k$$

for every $k \in \mathcal{K}$. The result follows. □

**Lemma 2.5.7.** *Let $\{||\cdot||\}_{k\in\mathcal{K}}$ be a family of uniformly equivalent norms, i.e., there exist positive constants $c_1$ and $c_2$ such that, for each $i, j \in \mathcal{K}$ and $x \in \mathbb{R}^n$, $c_1||x||_i \leq ||x||_j \leq c_2||x||_i$. Then $\{||\cdot||\}_{k\in\mathcal{K}}$ is uniformly equivalent to every norm on $\mathbb{R}^n$.*

*Proof.* Fix $j \in \mathcal{K}$. Then, for each $k \in \mathcal{K}$,

$$c_1||x||_j \leq ||x||_k \leq c_2||x||_j.$$

Let $||\cdot||$ denote some other norm. As $\mathbb{R}^n$ is finite-dimensional, there exists positive constants $\kappa_1$ and $\kappa_2$

such that

$$\kappa_1||x|| \leq ||x||_j \leq \kappa_2||x||.$$

It follows that

$$\kappa_1 c_1||x|| \leq c_1||x||_j \leq ||x||_k \leq c_2||x||_j \leq \kappa_2 c_2||x||.$$

Thus, the proof is complete. □

The previous two lemmas demonstrate that a uniformly equivalent family of norms on $\mathbb{R}^n$ need not be defined with respect to a secondary norm. Henceforth, a uniformly equivalent family of norms shall be defined as a family of norms that are uniformly equivalent to any norm. Properties of the matrix norms that are induced by a uniformly equivalent family of norms can be inferred as well.

**Lemma 2.5.8.** *Let $\{||\cdot||_k\}_{k\in\mathcal{K}}$ be a family of uniformly equivalent norms, and let $\{||\cdot||_{k\star}\}_{k\in\mathcal{K}}$ be the corresponding dual family of uniformly equivalent norms. Then the family of matrix norms $\{||\cdot||_{k,k\star}\}_{k\in\mathcal{K}}$ is also a uniformly equivalent family of norms.*

*Proof.* By definition, for each $j, k \in \mathcal{K}$,

$$c_1||x||_k \leq ||x||_j \leq c_2||x||_k, \quad \text{and} \quad \frac{1}{c_2}||y||_{k\star} \leq ||y||_{j\star} \leq \frac{1}{c_1}||y||_{k\star}$$

for each $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$. Let $A \in \mathbb{R}^{n\times n}$. Then

$$||H||_{j,j\star} = \max_{x\neq 0} \frac{||Hx||_{j\star}}{||x||_j} \leq \frac{1}{c_1^2} \max_{x\neq 0} \frac{||Hx||_{k\star}}{||x||_k}$$

and

$$||H||_{k,k\star} = \max_{x\neq 0} \frac{||Hx||_{k\star}}{||x||_k} \leq c_2^2 \max_{x\neq 0} \frac{||Hx||_{j\star}}{||x||_j}.$$

The result follows. □

**Lemma 2.5.9.** *Let $\{||\cdot||_k\}_{k\in\mathcal{K}}$ be a family of uniformly equivalent norms, and let $\{||\cdot||_{k\star}\}_{k\in\mathcal{K}}$ be the corresponding dual family of uniformly equivalent norms. Then the family of matrix norms $\{||\cdot||\}_{k\star,k}$ is also a uniformly equivalent family of norms.*

*Proof.* The result follows from Lemma 2.5.8 and the fact that the dual norm of a dual norm is the original norm, i.e., $||x||_{\star\star} = ||x||$ for all $x \in \mathbb{R}^n$. □

Now, when analyzing Algorithm 2.2 with the bound given in Lemma 2.5.3, it will sometimes be necessary to compare the difference of various quantities at different iterations. In this case, it will not be obvious which norm out of a family of uniformly equivalent norms should be applied. For this reason, a new norm is constructed out of an existing uniformly equivalent family of norms to suit this purpose.

**Theorem 2.5.10.** *Let $\{||\cdot||_k\}_{k\in\mathcal{K}}$ be a uniformly equivalent family of norms with constants $c_1$ and $c_2$ such that $c_1||x||_k \leq ||x||_j \leq c_2||x||_k$ for all $j,k \in \mathcal{K}$ and $x \in \mathbb{R}^n$. Consider the function*

$$\psi(x) = \sup_{k\in\mathcal{K}} ||x||_k.$$

*Then $\psi(x)$ is a norm, and $||x||_k \leq \psi(x) \leq c_2||x||_k$ for all $x \in \mathbb{R}^n$ and $k \in \mathcal{K}$.*

*Proof.* First, it must be established that $\psi(x)$ is in fact a norm. Let $x \in \mathbb{R}^n$ be some vector. Then $\psi(x) = \sup_{k\in\mathcal{K}} ||x||_k \geq 0$, with equality if and only if $x = 0$. Next, consider $\psi(\alpha x) = \sup_{k\in\mathcal{K}} ||\alpha x||_k = \sup_{k\in\mathcal{K}} |\alpha|||x||_k = |\alpha| \sup_{k\in\mathcal{K}} ||x||_k = |\alpha|\psi(x)$. It remains to show that the triangle inequality holds.

$$\psi(x+y) = \sup_{k\in\mathcal{K}} ||x+y||_k \leq \sup_{k\in\mathcal{K}} ||x||_k + ||y||_k \leq \sup_{k\in\mathcal{K}} ||x||_k + \sup_{k\in\mathcal{K}} ||y||_k = \psi(x) + \psi(y).$$

Thus, $\psi(x)$ is a norm. Let $||x||_{\mathcal{K}} = \psi(x)$ for all $x \in \mathbb{R}^n$. Now, for the bounds, it clearly holds that $||x||_k \leq ||x||_{\mathcal{K}}$ for all $k \in \mathcal{K}$. On the other hand, let $j \in \mathcal{K}$ be fixed. Then $||x||_{\mathcal{K}} = \sup_{k\in\mathcal{K}} ||x||_k \leq c_2||x||_j$. As $j$ is arbitrary, the upper bound holds. □

The upper bound on $||\cdot||_{\mathcal{K}}$ is crucial, as it prevents $||x||_{\mathcal{K}}$ from being infinite for some finite vector $x$. Now, note that in the definition of uniform equivalence, $c_1$ can be taken to be $1/c_2$, as the roles of $j$ and $k$ can be freely swapped. Henceforth, reference is made only to a single constant $c = c_2$.

**Lemma 2.5.11.** *Let $\{||\cdot||_k\}_{k\in\mathcal{K}}$ be a uniformly equivalent family of norms with equivalence constant $c$. Then*

$$\frac{1}{c}||y||_{k\star} \leq ||y||_{\mathcal{K}\star} \leq ||y||_{k\star}$$

*for all $k \in \mathcal{K}$, where $||\cdot||_{\mathcal{K}\star}$ is the dual norm of $||\cdot||_{\mathcal{K}}$.*

*Proof.* The result follows from the definition of $||\cdot||_{\mathcal{K}\star}$ and Lemma 2.5.4. □

The above result demonstrates that $||\cdot||_{\mathcal{K}\star}$ is not identical to $\sup_{k\in\mathcal{K}} ||\cdot||_{k\star}$. It remains to establish the equivalence of the uniformly equivalent family of induced matrix norms and the matrix norm induced by $||\cdot||_{\mathcal{K}}$ and $||\cdot||_{\mathcal{K}\star}$.

**Lemma 2.5.12.** *Let $\{||\cdot||_k\}_{k\in\mathcal{K}}$ be a uniformly equivalent family of norms with equivalence constant $c$. Then*

$$\frac{1}{c^2}||H||_{k,k\star} \leq ||H||_{\mathcal{K},\mathcal{K}\star} \leq ||H||_{k,k\star}$$

*for all $k \in \mathcal{K}$ and $H \in \mathbb{R}^{n\times n}$.*

*Proof.* Let $H \in \mathbb{R}^{n\times n}$, and let $k \in \mathcal{K}$. Then

$$||H||_{\mathcal{K},\mathcal{K}\star} = \max_{x\neq 0} \frac{||Hx||_{\mathcal{K}\star}}{||x||_{\mathcal{K}}} \leq \max_{x\neq 0} \frac{||Hx||_{k\star}}{||x||_k} = ||H||_{k,k\star},$$

and

$$||H||_{k,k\star} = \max_{x\neq 0} \frac{||Hx||_{k\star}}{||x||_k} \leq c^2 \max_{x\neq 0} \frac{||Hx||_{\mathcal{K}\star}}{||x||_{\mathcal{K}}} = c^2||H||_{\mathcal{K},\mathcal{K}\star},$$

□

Algorithm 2.2 can now be analyzed. In what follows, the model function $q_k$ is defined to be the quadratic function

$$q_k(x_k + p) = f(x_k) + g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H_k p,$$

where $g_k = \nabla f(x_k)$, but $H_k$ is not necessarily identical to $\nabla^2 f(x_k)$. Note then that $f(x_k) = q_k(x_k)$ and $\nabla f(x_k) = \nabla q_k(x_k)$. Let $\mathcal{S}$ and $\mathcal{V}$ denote the following iteration index sets:

$$\mathcal{S} = \{k : \rho_k \geq \eta_A\}, \quad \text{and} \quad \mathcal{V} = \{k : \rho_k \geq \eta_E\}.$$

These two definitions constitute the sets of successful and very successful iterations, respectively. The following properties are additionally assumed:

**Assumption 2.5.1.** *The objective function $f$ is twice continuously differentiable,*

**Assumption 2.5.2.** *The objective function $f$ is bounded below by some $\bar{f}$,*

**Assumption 2.5.3.** *The Hessian of the objective function is uniformly bounded, that is, there exists a constant $M_1$ such that $\{||\nabla^2 f(x)||'_k\} \leq M_1$ for all $k \in \mathcal{S}$, and*

**Assumption 2.5.4.** *The sequence of Hessians of the model functions is uniformly bounded, that is, there exists a constant $M_2$ such that $\{||H_k||'_k\} \leq M_2$ for all $k \in \mathcal{S}$.*

Assumption 2.5.3 and 2.5.4 and Lemma 2.5.12 imply that $\{||\nabla^2 f(x_k)||'_{\mathcal{S}}\} \leq M_1$ and $\{||H_k||'_{\mathcal{S}}\} \leq M_2$. Note that Assumption 2.5.3 is often too strong of an assumption. Without loss of generality, this

assumption can be replaced by the much weaker assumption that the Hessian is bounded for values of $x$ between two subsequent iterations. This is clearly satisfied when the level set $\mathcal{L}(f(x_0))$ is bounded. The following assumptions about the family of trust-region norms $||\cdot||_k$ used in Algorithm 2.2 must also be made:

**Assumption 2.5.5.** *The family of norms $\{||\cdot||_k\}$ is a uniformly equivalent family of norms with an equivalence constant $c$.*

This assumption is crucial, otherwise, in the limit, the norms may asymptotically "flatten out." From here on out, the proofs diverge from the standard presentation of the convergence results of Algorithm 2.2, in that the Cauchy point along a steepest-descent direction $h_k$ induced, at each iteration, by $g_k$ and a norm $||\cdot||'_k$ not necessarily identical to either $||\cdot||_k$ or $||\cdot||_2$. The following assumption on the family $\{||\cdot||'_k\}$ is made:

**Assumption 2.5.6.** *The family of norms $\{||\cdot||'_k\}$ is a uniformly equivalent family of norms with an equivalence constant $c'$.*

Let $||\cdot||_{\mathcal{S}}$ and $||\cdot||'_{\mathcal{S}}$ denote the norms $||x||_{\mathcal{S}} = \sup_{k\in\mathcal{S}} ||x||_k$ and $||x||'_{\mathcal{S}} = \sup_{k\in\mathcal{S}} ||x||'_k$, respectively. As both families of norms are uniformly equivalent, there exists a positive constant $\kappa$ such that

$$\frac{1}{\kappa}||x||_k \leq ||x||'_k \leq \kappa||x||_k$$

for all $x \in \mathbb{R}^n$ and $k \in \mathcal{S}$.

In light of Lemma 2.5.3, the following assumption on the computed solution of the trust-region subproblem in Algorithm 2.2 is made:

**Assumption 2.5.7.** *For all iterations $k$, let $p_k$ be the computed approximate solution to the $k$th trust-region subproblem. Assume that*

$$q_k(x_k) - q_k(x_k + p_k) \geq \tau_1||g_k||'_{k\star} \min\{\delta_k, \frac{||g_k||'_{k\star}}{||H_k||'_k}\}, \tag{2.21}$$

*where $||H||'_k = \max_{x\neq 0} ||Hx||'_{k\star}||x||_k$, and $h_k$ is the steepest-descent direction induced by $||\cdot||'_k$ satisfying $||h_k||'_k = ||g_k||'_{k\star}$, for some $\tau_1 \in (0,1)$.*

This implies that the model decrease is at least as large as a fraction of the model decrease had the Cauchy step $p_k^C$ been taken instead of $p_k$. In practice, $p_k$ will be much closer to the true solution $p_k^\star$, but for theoretical purposes, this bound is sufficient to prove convergence. These assumptions are the

assumptions made in [7] Chapter 6 to analyze the basic trust-region method. The following result follows directly from Assumption 2.5.7 .

**Lemma 2.5.13.** *If Assumption 2.5.7 holds, and $\nabla f(x_k) \neq 0$, then $q_k(x_k + p_k) < q_k(x_k)$ and $p_k \neq 0$.*

The value of $\rho_k$ is then well-defined as long as $x_k$ is not a first-order stationary point. Assumption 2.5.7 can be shown to hold in cases where the trust-region subproblem is assumed to be solved to a high degree of accuracy, as seen in the following result.

**Theorem 2.5.14** ([7] Chapter 6)**.** *Suppose that for all $k$, the computed solution $p_k$ satisfies*

$$q_k(x_k) - q(x_k + p_k) \geq \bar{\tau}_1(q_k(x_k) - q_k(x_k + p_k^\star)),$$

*where $\bar{\tau}_1 \in (0, 1)$. Then Assumption 2.5.7 holds for some value of $\tau_1$.*

*Proof.* It trivially holds that $q_k(x_k + p_k^\star) \leq q_k(x_k + p_k^C)$. Then

$$q_k(x_k) - q(x_k + p_k) \geq \bar{\tau}_1(q_k(x_k) - q_k(x_k + p_k^\star)) \geq \bar{\tau}_1(q_k(x_k) - q_k(x_k + p_k^C)).$$

The result then follows from Lemma 2.5.3. □

This result demonstrates the value of assuming the model is at least as good as a fraction of the Cauchy point, as more practical implementations fall within this framework.

**First-order Convergence**

The goal of this section is to demonstrate that Algorithm 2.2 generates a sequence of point $\{x_k\}$ such that all limit points satisfy the first-order necessary conditions for optimality. The results an theorems of this section closely follow those of [7] Chapter 6, with some minor technical results changed to reflect the use of a different definition of the Cauchy step. This first result bounds the error between $f$ and the model function $q_k$ at the new iterates $x_k + p_k$.

**Theorem 2.5.15.** *Suppose that Assumptions 2.5.1, 2.5.3, 2.5.4, 2.5.5, and 2.5.6 hold. Then, for all $k$,*

$$|f(x_k + p_k) - q_k(x_k + p_k)| \leq \left(\frac{||p_k||'_k}{||p||_k}\right)^2 \max\{M_1, M_2\}\delta_k^2,$$

*where $M_1$ is the upper bound of the sequence $\{||\nabla^2 f(x_k)||'_k\}$ and $M_2$ is the upper bound of the sequence $\{||H_k||'_k\}$.*

*Proof.* By the mean value theorem, it holds that

$$f(x_k + p_k) = f(x_k) + \nabla f(x_k)^{\mathrm{T}} p_k + \frac{1}{2} p_k^{\mathrm{T}} \nabla^2 f(x_k + \xi_k p_k) p_k$$

for some $\xi_k \in [0, 1]$. The definition of $q_k$ gives

$$q_k(x_k + p_k) = f(x_k) + \nabla f(x_k)^{\mathrm{T}} p_k + \frac{1}{2} p_k^{\mathrm{T}} H_k p_k.$$

Subtracting these two equations and taking the absolute value gives

$$\begin{aligned}
|f(x_k + p_k) - q_k(x_k + p_k)| &= \frac{1}{2} |p_k^{\mathrm{T}} \nabla^2 f(x_k + \xi_k p_k) p_k - p_k^{\mathrm{T}} H_k p_k| \\
&\leq \frac{1}{2} |p_k^{\mathrm{T}} \nabla^2 f(x_k + \xi_k p_k) p_k| + \frac{1}{2} |p_k^{\mathrm{T}} H_k p_k| \\
&\leq \frac{1}{2} (M_1 + M_2) (||p_k||'_k)^2 \\
&\leq \frac{1}{2} (M_1 + M_2) (||p_k||_k)^2 \frac{(||p_k||'_k)^2}{(||p_k||_k)^2} \\
&\leq \max\{M_1, M_2\} \delta_k^2 \frac{(||p_k||'_k)^2}{(||p_k||_k)^2}.
\end{aligned}$$

□

**Corollary 2.5.15.1.** *If 2.5.5 and 2.5.6 hold, then there exists a constant $\tau_2$ such that*

$$|f(x_k + p_k) - q_k(x_k + p_k)| \leq \tau_2 \delta_k^2, \tag{2.22}$$

*with*

$$\tau_2 = \kappa^2 \max\{M_1, M_2\}.$$

Thus, the error between the objective function and the model function decreases quadratically with the trust-region radius. This intuitively leads to the next result, which states that if the trust-region radius is sufficiently small, then the next iteration is guaranteed to be very successful.

**Theorem 2.5.16.** *Suppose that Assumptions 2.5.1, 2.5.3, 2.5.4, 2.5.5, 2.5.6, and 2.5.7 all hold. Suppose that $g_k \neq 0$, and that*

$$\delta_k \leq \frac{\tau_1 ||g_k||'_{k\star} (1 - \eta_E)}{\tau_2}. \tag{2.23}$$

*Then iteration $k$ is very successful.*

*Proof.* Recall that $\tau_1 \in (0,1)$ and $\eta_E \in (0,1)$. Thus $\tau_1(1-\eta_E) < 1$. Conditions (2.23) and (2.22) then imply that

$$\delta_k < \frac{||g_k||_{k\star}}{M_1}. \tag{2.24}$$

Assumption 2.5.7 then gives

$$q_k(x_k) - q_k(x_k + p_k) \geq \tau_1 ||g_k||'_{k\star} \delta_k. \tag{2.25}$$

On the other hand, Theorem 2.5.15, Corollary 2.5.15.1, (2.24), (2.25), and (2.23) yield

$$|\rho_k - 1| = \left| \frac{f(x_k + p_k) - q_k(x_k + p_k)}{q_k(x_k) - q_k(x_k + p_k)} \right| \leq \frac{\tau_2}{\tau_1 ||g_k||'_{k\star}} \delta_k \leq 1 - \eta_E.$$

Therefore, $\rho_k \geq \eta_E$, and the iteration is very successful. □

It then follows logically that the trust-region radius cannot become too small. This property shows that progress will always be made unless the algorithm arrives at a critical point.

**Theorem 2.5.17.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, and 2.5.7 hold. Further suppose there exists a constant $\bar{g}$ such that $||g_k||'_{k\star} \geq \bar{g}$ for all iterations $k$. Then there is a constant $\bar{\delta}$ such that $\delta_k \geq \bar{\delta}$ for all iterations $k$.*

*Proof.* Let $k$ denote the first iteration at which

$$\delta_{k+1} \leq \frac{\gamma_C \tau_1 \bar{g}(1-\eta_E)}{\tau_2} \tag{2.26}$$

holds. As $k$ is the first such iterate, it must hold that $\gamma_C \delta_k \leq \delta_{k+1}$, and thus

$$\delta_k \leq \frac{\tau_1 \bar{g}(1-\eta_E)}{\tau_2}.$$

Therefore, (2.23) is satisfied, and iteration $k$ is very successful. But this contradicts that $k$ was the first such index that (2.26) holds. It follows that no iteration satisfies (2.26), and thus the trust-region radii are bounded below when the norm of $g_k$ is bounded below. □

Now, consider the case where the number of successful iterations is finite. The following result holds.

**Theorem 2.5.18.** *Suppose that assumptions 2.5.1, 2.5.3, 2.5.4, 2.5.5, 2.5.6, and 2.5.7 hold, and that the number of successful iterations is finite. Then $x_k = x^*$ for all $k$ sufficiently large, and $x^*$ is a first-order*

*stationary point.*

*Proof.* Algorithm 2.2 ensures that $x^* = x_{k+1} = x_{k+j}$ for all $j > 0$ if $k$ is the last successful iteration. Now, all iterations are unsuccessful for sufficiently large $k$. Therefore $\lim_{k\to\infty} \delta_k = 0$. If $||g_{k+1}||'_{k+1\star} > 0$, then Theorem 2.5.16 implies that there must be a successful iteration after iteration $k$. This is a contradiction. Thus $||g_{k+1}||'_k = 0$, and Assumption 2.5.6 implies that $x^*$ is a first-order stationary point. □

The more likely scenario is that there will be infinitely many successful iterations. The following result established that the algorithm converges to a stationary point on a subsequence of the iterations.

**Theorem 2.5.19.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, and 2.5.7 hold. Then*

$$\liminf_{k\to\infty} ||\nabla f(x_k)||'_{k\star} = 0.$$

*Proof.* For the sake of establishing a contradiction, assume that the result does not hold, i.e., there exists an $\varepsilon > 0$ such that $||\nabla f(x_k)||'_{k\star} = ||g_k||'_{k\star} \geq \varepsilon$ for all $k$. Suppose now that $k$ is a successful iteration. It then follows that

$$f(x_k) - f(x_k + p_k) \geq \eta_A(q_k(x_k) - q_k(x_k + p_k)) \geq \tau_1 \varepsilon \eta_A \min\{\frac{\varepsilon}{M_2}, \bar{\delta}\}.$$

Let $n_k$ denote the number of successful iterations in between the initial iteration and the $k$th iteration. Then

$$f(x_0) - f(x_{k+1}) = \sum_{j=0, j\in\mathcal{S}}^{k} (f(x_j) - f(x_{j+1})) \geq n_k \tau_1 \varepsilon \eta_A \min\{\frac{\varepsilon}{M_2}, \bar{\delta}\}.$$

Now, as there are infinitely many successful iterations, $n_k \to \infty$ as $k \to \infty$. Thus, the difference between $f(x_0)$ and $f(x_{k+1})$ is unbounded. This contradicts Assumption 2.5.2. The result follows. □

Theorem 2.5.19 can be leveraged to prove the following stronger result.

**Theorem 2.5.20.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, and 2.5.7 hold. Then*

$$\lim_{k\to\infty} ||\nabla f(x_k)||'_{k\star} = 0.$$

*Proof.* Assume, for the sake of establishing a contradiction, that the result does not hold, i.e., that there exists a subsequence of successful iterates such that $||\nabla f(x_k)||'_{k\star} \geq ||\nabla f(x_k)||'_{\mathcal{S}\star} \geq 2\varepsilon$ for some $\varepsilon > 0$. Theorem 2.5.19 and Assumption 2.5.6 ensures that for each iteration $k$, there exists a subsequent iteration

$\ell_k$ such that $\ell_k$ is the first iteration larger than $k$ to satisfy $||g_{\ell_k}||'_{\mathcal{S}\star} \leq ||g_{\ell_k}||'_{\ell_k\star} < \varepsilon$. Let $\mathcal{K}$ denote the subset of successful iterations from $k$ to $\ell_k$, i.e.,

$$\mathcal{K} = \{j \in \mathcal{S} : k \leq j \leq \ell_k\}.$$

Then, it holds that, for all $k \in \mathcal{K}$,

$$f(x_k) - f(x_{k+1}) \geq \eta_A(q_k(x_k) - q_k(x_k + p_k)) \geq \tau_1 \varepsilon \eta_A \min\{\frac{\varepsilon}{M_2}, \delta_k\}. \tag{2.27}$$

Now, the sequence $\{f(x_k)\}$ is monotonically decreasing and bounded below, so $f(x_k) - f(x_{k+1}) \to 0$, and therefore $\delta_k \to 0$. It follows then that, for $k \in \mathcal{K}$ sufficiently large,

$$\delta_k \leq \frac{1}{\tau_1 \varepsilon \eta_A}(f(x_k) - f(x_{k+1})). \tag{2.28}$$

From Assumption 2.5.5 and 2.5.6, it follows that,

$$\begin{aligned} ||x_k - x_{\ell_k}||'_{\mathcal{S}} &\leq \sum_{i=k, i\in\mathcal{K}}^{\ell_k} ||x_i - x_{i+1}||'_{\mathcal{S}} \\ &\leq c' \sum_{i=k, i\in\mathcal{K}}^{\ell_k} ||x_i - x_{i+1}||'_i \\ &\leq c' \sum_{i=k, i\in\mathcal{K}}^{\ell_k} \frac{||x_i - x_{i+1}||'_i}{||x_i - x_{i+1}||_i} \delta_i \\ &\leq \frac{c'\kappa}{\tau_1 \varepsilon \eta_A}(f(x_k) - f(x_{\ell_k})). \end{aligned} \tag{2.29}$$

The right-hand side of this inequality must converge to zero, and thus $||x_k - x_{\ell_k}||'_{\mathcal{S}}$ converges to zero. By the continuity of the gradient, it follows that $||\nabla f(x_k) - \nabla f(x_{\ell_k})||'_{\mathcal{S}\star}$ converges to zero. However, the definitions of $k$ and $\ell_k$ ensure that $||g_k - g_{\ell_k}||'_{\mathcal{S}\star} \geq \varepsilon$. This is a contradiction. The result follows. □

This powerful result demonstrates that all limit points of the sequence of iterates satisfy the first-order necessary conditions. In the next section, similar results are established for second-order optimality conditions.

**Second-order Convergence**

Up until this point, the model functions $q_k$ have been assumed to agree with $f$ at $x_k$ up to first-order. Outside of boundedness, no assumptions on the matrices $H_k$ have been made. In order to

ensure convergence to points that satisfy second-order conditions, the second-order information of the model $q_k$ must be considered. As with the previous section, the results and theorems of the previous section closely follow those of [7] Chapter 6, with some technical details changed to reflect the use of a different definition of the Cauchy step.

Notice that Theorem 2.5.20 only guarantees that $\{g_k\}$ converges to zero. It does not guarantee that the sequence of iterates $x_k$ converges to a single limit point. In order to show this result, second-order information is needed. The second-order derivatives of the model provide insight into the curvature of the model. The following defines the notion of the *least curvature* of the quadratic model $q_k$.

**Definition 2.5.5.** The *least curvature* of the quadratic model $q_k$ in the $||\cdot||'_k$ norm is defined as

$$\lambda'_k(H_k) = \min_{||x||'_k=1} x^{\mathrm{T}} H_k x.$$

Note that if $||\cdot||'_k$ is the 2-norm, then $\lambda'_k(H_k) = \lambda_{\min}(H_k)$, and if $||\cdot||'_k$ is the $B_k$-norm for some positive definite matrix $B_k$, then $\lambda'_k(H_k) = \lambda_{\min}(H_k, B_k)$. Here again, the results depart from the standard presentation, where $\lambda'_k$ is always taken to be $\lambda_{\min}(H_k)$. The curvature in the $||\cdot||'_{\mathcal{S}}$ norm is similarly defined as

$$\lambda'_{\mathcal{S}}(H_k) = \min_{||x||'_{\mathcal{S}}=1} x^{\mathrm{T}} H_k x.$$

Note that if $H_k$ is positive semidefinite, then

$$\begin{aligned} \lambda'_{\mathcal{S}}(H_k) &= \min_{||x||'_{\mathcal{S}}=1} x^{\mathrm{T}} H_k x \\ &= \min_{x\neq 0} \frac{x^{\mathrm{T}} H_k x}{(||x||'_{\mathcal{S}})^2} \\ &\leq \min_{x\neq 0} \frac{x^{\mathrm{T}} H_k x}{(||x||'_k)^2} = \min_{||x||'_k=1} x^{\mathrm{T}} H_k x = \lambda'_k(H_k) \end{aligned} \tag{2.30}$$

**Lemma 2.5.21.** *Suppose that $\lambda'_k(H_k) \geq \varepsilon > 0$. Then*

$$||p_k||'_k \leq \frac{2}{\varepsilon} ||g_k||'_{k\star}.$$

*Proof.* Consider the model decrease at $x_k + p_k$, given by

$$q_k(x_k) - q_k(x_k + p_k) = -g^{\mathrm{T}} p_k - \frac{1}{2} p_k^{\mathrm{T}} H_k p_k.$$

Two cases are considered. First, consider the case where $q_k(x_k) = q_k(x_k + p_k)$. Then $g_k = 0$. Therefore,

$p_k^{\mathrm{T}} H_k p_k = 0$. But, by the fact that $\lambda_k'(H_k) \geq \varepsilon$, $H_k$ is positive definite, so $p_k = 0$, in which case the result holds trivially. Now, consider the more common case where $q_k(x_k + p_k) < q_k(x_k)$, and $p_k \neq 0$. Let

$$\phi(t) = q_k(x_k) - q_k(x_k + tp_k) = -tg_k^{\mathrm{T}} p_k - \frac{t^2}{2} p_k^{\mathrm{T}} H_k p_k$$

for positive values of $t$. By assumption, $\phi$ is concave and quadratic, i.e., a downwards-facing parabola. Note that $\phi(0) = 0$ and $\phi(1) > 0$ by construction. Furthermore, note that $g_k^{\mathrm{T}} p_k < 0$. Then the maximum of $\phi$ satisfies

$$t_\star = \operatorname*{argmax}_t \phi(t) \geq \frac{1}{2},$$

Furthermore, a simple calculation yields

$$t_\star = \frac{|g_k^{\mathrm{T}} p_k|}{p_k^{\mathrm{T}} H_k p_k} \leq \frac{||g_k||_{k\star}'}{\varepsilon ||p_k||_k'}.$$

Combining these two equations yields the result. □

**Corollary 2.5.21.1.** *Suppose that $\lambda_{\mathcal{S}}'(H_k) \geq \varepsilon > 0$. Then*

$$||p_k||_k' \leq ||p_k||_{\mathcal{S}}' \leq \frac{2}{\varepsilon} ||g_k||_{\mathcal{S}\star}' \leq \frac{2}{\varepsilon} ||g_k||_{k\star}'.$$

*Proof.* The proof is identical to the proof of Lemma 2.5.21 □

The behavior of Algorithm 2.2 is now examined when the sequence of approximate Hessian $H_k$ are asymptotically positive definite along a subsequence of iterates.

**Theorem 2.5.22.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, and 2.5.7 hold. Further, suppose that $\{x_{k_i}\}$ is a subsequence of iterates converging to the first-order critical point $x^*$ and that there exists a positive constant $\widehat{\lambda}$ such that $\lambda_k'(H_k) \geq \lambda_{\mathcal{S}}'(H_k) \geq \widehat{\lambda}$ for all $k$ such that $x_k$ is sufficiently close to $x^*$. Furthermore, suppose that $\nabla^2 f(x^*)$ is nonsingular. Then the sequence $\{x_k\}$ converges to $x^*$.*

*Proof.* By Theorem 2.5.20, $x^*$ is a first-order stationary point. If there are only finitely many successful iterations, then the result follows from Theorem 2.5.18. Assume without loss of generality that the subsequence $\{x_{k_i}\}$ consists only of successful iterations, so that

$$x_{k_i+1} = x_{k_i} + p_{k_i}$$

for all $i$. Let $\delta > 0$ be a constant so that $\lambda'_k(H_k) \geq \lambda'_{\mathcal{S}}(H_k) \geq \widehat{\lambda}$ holds for all $k$ such that $||x_k - x^*||'_k \leq ||x_k - x^*||'_{\mathcal{S}} \leq \delta$ and that

$$||\nabla^2 f(x) - \nabla^2 f(x^*)||'_{\mathcal{S}} \leq \frac{1}{4}\min\{1, \widehat{\lambda}, \sigma'_{\mathcal{S}}(\nabla^2 f(x^*))\} := d_0 \tag{2.31}$$

for all $x$ such that $||x - x^*||'_{\mathcal{S}} \leq \delta$, where $\sigma'_{\mathcal{S}}(\nabla^2 f(x^*)) = \min_{||x||'_{\mathcal{S}}=1} ||\nabla^2 f(x^*)x||'_{\mathcal{S}\star}$, i.e., the measure of the curvature of $\nabla^2 f(x^*)$ of least magnitude. As $\nabla^2 f(x^*)$ is assumed to be nonsingular, it holds that $\sigma'_{\mathcal{S}}(\nabla^2 f(x^*)) > 0$. Let $i_1$ be an index large enough to ensure

$$||x_{k_i} - x^*||'_{\mathcal{S}} \leq \frac{\widehat{\lambda}\delta}{2d_0 + \widehat{\lambda}} := d_1 \tag{2.32}$$

for all $i \geq i_1$ and that

$$||g_k||'_{\mathcal{S}\star} \leq ||g_k||'_{k\star} \leq d_0 d_1 < \delta \tag{2.33}$$

for all $k \geq k_{i_1}$. Equation (2.32) is possible due to the convergence of $\{x_{k_i}\}$ to $x^*$, while (2.33) is possible because of Theorem 2.5.20. Lemma 2.5.21 may now be applied at iteration $k_i$ with $\varepsilon = \widehat{\lambda}$ to yield

$$||p_{k_i}||'_{k_i} \leq ||p_{k_i}||'_{\mathcal{S}} \leq \frac{2}{\widehat{\lambda}}||g_{k_i}||'_{\mathcal{S}\star} \leq \frac{2}{\widehat{\lambda}}||g_{k_i}||'_{k\star}.$$

Thus, it follows that

$$||x_{k_i+1} - x^*||'_{\mathcal{S}} \leq ||x_{k_i} - x^*||'_{\mathcal{S}} + ||p_k||'_{\mathcal{S}} \leq \left(1 + \frac{2d_0}{\widehat{\lambda}}\right) d_1 = \delta. \tag{2.34}$$

Next, assume that

$$||x_{k_i+1} - x^*||'_{\mathcal{S}} > d_1. \tag{2.35}$$

See now that

$$g_{k_i+1} = \nabla f(x_{k_i+1}) = \nabla f(x^*) + \int_0^1 \nabla^2 f(x_{k_i+1} + t(x^* - x_{k_i+1}))(x_{k_i+1} - x^*)dt.$$

By the triangle inequality, the definition of $\sigma'_{\mathcal{S}}(\nabla^2 f(x^*))$, (2.34), the fact that $\nabla f(x^*) = 0$, and (2.35),

$$\begin{aligned}
||g_{k_i+1}||'_{\mathcal{S}\star} &= ||\nabla^2 f(x^*)(x_{k_i+1} - x^*) + \int_0^1 \big(\nabla^2 f(x_{k_i+1} + t(x^* - x_{k_i+1})) - \nabla^2 f(x^*)\big)(x_{k_i+1} - x^*)dt||'_{\mathcal{S}\star} \\
&\geq ||\nabla^2 f(x^*)(x_{k_i+1} - x^*)||'_{\mathcal{S}\star} - \left|\left| \int_0^1 \big(\nabla^2 f(x_{k_i+1} + t(x^* - x_{k_i+1})) - \nabla^2 f(x^*)\big)dt \right|\right|'_{\mathcal{S}} ||x_{k_i+1} - x^*||'_{\mathcal{S}} \\
&> \sigma'_{\mathcal{S}}(\nabla^2 f(x^*))d_1 - \left|\left| \int_0^1 \big(\nabla^2 f(x_{k_i+1} + t(x^* - x_{k_i+1})) - \nabla^2 f(x^*)\big)dt \right|\right|'_{\mathcal{S}} \delta.
\end{aligned} \tag{2.36}$$

Additionally,

$$\begin{aligned}
&\left|\left| \int_0^1 \big(\nabla^2 f(x_{k_i+1} + t(x^* - x_{k_i+1})) - \nabla^2 f(x^*)\big)dt \right|\right|'_{\mathcal{S}} \\
&\qquad \leq \max_{t\in[0,1]} ||\nabla^2 f(x_{k_i+1} + t(x^* - x_{k_i+1})) - \nabla^2 f(x^*)||'_{\mathcal{S}} \\
&\qquad \leq d_0.
\end{aligned} \tag{2.37}$$

By combining (2.36), (2.37), and the definitions of $d_0$ and $d_1$, it holds that

$$||g_{k_i+1}||'_{\mathcal{S}\star} > \sigma'_{\mathcal{S}}(\nabla^2 f(x^*))d_1 - d_0\delta \geq 4d_0d_1 - d_0d_1\frac{2d_0 + \widehat{\lambda}}{\widehat{\lambda}} = \frac{d_0d_1(3\widehat{\lambda} - 2d_0)}{\widehat{\lambda}} > d_0d_1.$$

This contradicts (2.33). Therefore,

$$||x_{k_i+1} - x^*||'_{\mathcal{S}} \leq d_1 < \delta.$$

All the conditions established at $x_{k_i}$ remain satisfied at $x_{k_i+1}$. Thus, for all $j > 1$, it holds that

$$||x_{k_i+j} - x^*||'_{\mathcal{S}} \leq d_1 < \delta.$$

Taking the limit $\delta \to 0$ shows that $\{x_k\}$ converges to $x^*$. □

An additional assumption of $H_k$ is added to ensure that the model asymptotically coincides with the objective up to the second-order terms.

**Assumption 2.5.8.** *Assume that* $\lim_{k\to\infty} ||\nabla^2 f(x_k) - H_k||'_k = 0$ *whenever* $\lim_{k\to\infty} ||g_k||'_k = 0$.

Two technical lemmas are now presented, the second of which shall be used to improve the convergence results.

**Lemma 2.5.23.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, 2.5.7, and 2.5.8 hold. Further suppose that there exists a subsequence* $\{k_i\}$ *and a constant* $m > 0$ *such that*

$$q_{k_i}(x_{k_i}) - q_{k_i}(x_{k_i} + p_{k_i}) \geq m(||p_{k_i}||'_{k_i})^2 > 0$$

*for all $i$ sufficiently large. Furthermore, suppose that* $\lim_{i\to\infty} ||p_{k_i}||_{k_i} = 0$. *Then* $\rho_{k_i} \geq \eta_E$ *for $i$ sufficiently large.*

*Proof.* The mean value theorems implies that for $i$ sufficiently large and some constant $\xi_{k_i} \in (0,1)$,

$$\begin{aligned}
|\rho_{k_i} - 1| &= \left| \frac{f(x_{k_i} + p_{k_i}) - q_{k_i}(x_{k_i} + p_{k_i})}{q_{k_i}(x_{k_i}) - q_{k_i}(x_{k_i} + p_{k_i})} \right| \\
&\leq \frac{1}{m(||p_{k_i}||'_{k_i})^2} |p_k^{\mathrm{T}} \nabla^2 f(x_{k_i} + \xi_{k_i} p_{k_i}) p_k - p_k^{\mathrm{T}} H_k p_k| \\
&= \frac{1}{m(||p_{k_i}||'_{k_i})^2} |p_k^{\mathrm{T}} (\nabla^2 f(x_{k_i} + \xi_{k_i} p_{k_i}) - H_k) p_k| \\
&\leq \frac{1}{m} ||\nabla^2 f(x_{k_i} + \xi_{k_i} p_{k_i}) - H_k||'_{k_i} \\
&= \frac{1}{m} ||(\nabla^2 f(x_{k_i} + \xi_{k_i} p_{k_i}) - \nabla^2 f(x_{k_i})) - (H_k - \nabla^2 f(x_{k_i}))||'_{k_i} \\
&\leq \frac{1}{m} \left( ||\nabla^2 f(x_{k_i} + \xi_{k_i} p_{k_i}) - \nabla^2 f(x_{k_i})||_{k_i} + ||H_k - \nabla^2 f(x_{k_i})||'_{k_i} \right).
\end{aligned}$$

Now, both terms on the left-hand side converge to zero as $i$ goes to infinity. and thus $\rho_{k_i}$ converges to 1. The result follows. □

A consequence of Lemma 2.5.23 is the following:

**Lemma 2.5.24.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, 2.5.7, and 2.5.8 hold. Further suppose that there exists a positive constant $m$ such that*

$$q_k(x_k) - q_k(x_k + p_k) \geq m(||p_k||'_k)^2$$

*for all $k$ sufficiently large. Additionally, suppose that* $\lim_{k\to\infty} ||p_k||'_k = 0$. *Then all iterations are eventually very successful and $\delta_k$ is bounded away from zero.*

The next result uses this lemma to show that if one of the limit points of the sequence of iterates is an isolated minimizer, then the full sequence converges to the said minimizer. Furthermore, the steps $p_k$ eventually do not depend on the trust-region radius.

**Theorem 2.5.25.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, 2.5.7, and 2.5.8 hold, and that $\{x_{k_i}\}$ is a subsequence of iterates generated by Algorithm 2.2 converging to the first-order stationary point $x^*$. Suppose that the steps $p_k \neq 0$ for $k$ are sufficiently large. Finally, suppose that $x^*$ satisfies second-order sufficient optimality conditions, i.e., that $\nabla^2 f(x^*) \succ 0$. The $\{x_k\}$ converges to $x^*$, all iterations are eventually very successful, and the trust-region radius $\delta_k$ is bounded away from zero.*

*Proof.* Theorem 2.5.20, Assumption 2.5.8, and the positive definiteness of $\nabla^2 f(x^*)$ imply that $H_{k_i}$ is positive definite for $i$ sufficiently large, where the subsequence $\{k_i\}$ is chosen to be the set of successful iteration indices converging to $x^*$. Thus, there exists a constant $\widehat{\lambda}$ such that $\lambda'_k(H_k) \geq \lambda'_{\mathcal{S}}(H_k) \geq \widehat{\lambda}$ holds for any such subsequence. Theorem 2.5.22 can then be applied to see that $\{x_k\}$ converges to $x^*$. Lemma 2.5.21 can be applied to show that

$$||g_k||'_k \geq ||g_k||'_{\mathcal{S}\star} \geq \frac{\widehat{\lambda}}{2}||p_k||'_{\mathcal{S}} \geq \frac{\widehat{\lambda}}{2}||p_k||'_k > 0 \tag{2.38}$$

for $k$ sufficiently large. Assumption 2.5.7 and the fact that $||p_k||'_k = ||p_k||_k \frac{||p_k||'_k}{||p_k||_k} \leq \delta_k \frac{||p_k||'_k}{||p_k||_k} \leq \kappa\delta_k$ yield

$$\begin{aligned} q_k(x_k) - q_k(x_k + p_k) &\geq \frac{1}{2}\widehat{\lambda}\tau_1 ||p_k||'_k \min\{\frac{\widehat{\lambda}||p_k||'_k}{2M_2}, \delta_k\} \\ &\geq \frac{1}{2}\widehat{\lambda}\tau_1 ||p_k||'^2_k \min\{\frac{\widehat{\lambda}}{2M_2}, \frac{1}{\kappa}\} \\ &\geq \delta ||p_k||'^2_k \end{aligned} \tag{2.39}$$

where

$$\delta = \frac{1}{2}\widehat{\lambda}\tau_1 \min\{\frac{\widehat{\lambda}}{2M_2}, \frac{1}{\kappa}\}.$$

As $g_k \to 0$ by Theorem 2.5.20, (2.38) additionally also shows that

$$\lim_{k\to\infty} ||p_k||'_k = 0.$$

Lemma 2.5.24 can then be applied, and the result follows. □

Theorem 2.5.25 fully captures what occurs when the algorithm converges to an isolated minimizer. Next, the convergence of the sequence of iterates to points that satisfy second-order necessary conditions is examined without assuming the positive definiteness of the Hessian of the objective at the limit points. Such points may be minimizers, but this is not necessarily the case. Such a result can only be possible if the algorithm is able to avoid converging to maximizers and saddle points.

So far, it has only been assumed that the step $p_k$ yields a model reduction at least as large as the predicted reduction yielded by the Cauchy step. The predicted reduction when the step $p_k$ takes advantage of directions of negative curvature is now examined. Consider the model Hessian $H_k$ at iteration $k$. Assume that it is indefinite, i.e., has at least one direction of negative curvature. Here again, the proofs diverge from the standard presentation by considering directions of negative curvature with respect to

the norm $||\cdot||'_k$, as opposed to $||\cdot||_2$. Let $\theta_k = \lambda'_k(H_k) = \min_{||x||'_k=1} x^{\mathrm{T}} H_k x$. The assumption that $H_k$ is indefinite guarantees that $\theta_k < 0$. Let $u_k$ be a direction that achieves a fraction of this negative curvature, scaled so that $||u_k||_k = \delta_k$ and $g_k^{\mathrm{T}} u_k \leq 0$. Then

$$u_k^{\mathrm{T}} H_k u_k \leq \bar{\tau}\theta_k \delta_k^2 \frac{(||u_k||'_k)^2}{||u_k||_k^2} \tag{2.40}$$

for some constant $\bar{\tau} \in (0, 1]$. Just like with the Cauchy step, the goal is to minimize the trust-region subproblem along the vector $u_k$. By construction, the minimizer is simply $u_k$. An additional requirement that the overall model decrease at $x_k + p_k$ to satisfy

$$q_k(x_k) - q_k(x_k + p_k) \geq \tau_3(q_k(x_k) - \min\{q_k(x_k + p_k^C), q_k(x_k + u_k)) \tag{2.41}$$

for some $\tau_3 \in (0, 1]$ is enforced.

**Theorem 2.5.26.** *Suppose that $H_k$ is indefinite. Then*

$$q_k(x_k) - q_k(x_k + u_k) \geq -\frac{1}{2}\bar{\tau}\theta_k \delta_k^2 \frac{(||u_k||'_k)^2}{||u_k||_k^2}$$

*Proof.* The result follows from the definition of $q_k$ and (2.40). □

Putting all assumptions on the model decrease together gives

$$q_k(x_k) - q_k(x_k + p_k) \geq \tau_3 \max\left\{\frac{1}{2}||g_k||'_{k\star} \min\left\{\frac{||g_k||_{k\star}}{||H_k||'_k}, \delta_k \frac{||g_k||'_\star}{||h_k||_k}\right\}, -\frac{1}{2}\bar{\tau}\theta_k \delta_k^2 \frac{(||u_k||'_k)^2}{||u_k||_k^2}\right\}$$

This equation is quite cumbersome to work with. To simplify things, the following assumption on Algorithm 2.2 is made alongside Assumption 2.5.7:

**Assumption 2.5.9.** *If $\theta_k < 0$, then*

$$q_k(x_k) - q_k(x_k + p_k) \geq \tau_4 |\theta_k| \delta_k^2$$

*for some constant $\tau_4 \in (0, 1/2)$*

This assumption is essentially stating that if the model function has negative curvature, a first-order stationary point is approached, and if the second-order terms of the model appear to be relevant, then the negative curvature is not ignored by the computed trust-region step.

The first result with the new assumption states that the objective function must be convex on a neighborhood of a subsequence of the iterates.

**Theorem 2.5.27.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, 2.5.7, 2.5.8, and 2.5.9 hold. Then*

$$\limsup_{k\to\infty} x^{\mathrm{T}}\nabla^2 f(x_k)x \geq 0.$$

*for all* $x \in \mathbb{R}^n$,

*Proof.* For the sake of establishing a contradiction, assume that there exists a constant $\widehat{\lambda} < 0$ such that

$$\lambda'_k(\nabla^2 f(x_k)) = \min_{||x||'_k=1} x^{\mathrm{T}}\nabla^2 f(x_k)x \leq \widehat{\lambda} < 0.$$

Theorem 2.5.20 guarantees that $\lim_{k\to} ||g_k||'_{k\star} = 0$. Assumption 2.5.8 yields that, for $k$ sufficiently large,

$$\lambda'_k(H_k) \leq \frac{1}{2}\widehat{\lambda}.$$

Combining this with Assumption 2.5.9 gives

$$q_k(x_k) - q_k(x_k + p_k) \geq \frac{1}{2}\tau_4|\widehat{\lambda}|\delta_k^2.$$

Lemma 2.5.23 applied to the full sequence and the bound

$$||p_k||'_k = ||p_k||_k \frac{||p_k||'_k}{||p_k||_k} \leq \delta_k \frac{||p_k||'_k}{||p_k||_k} \leq \kappa\delta_k$$

shows that there exists an index $k_0$ and a scalar $d_1$ such that

$$\rho_k \geq \eta_{\scriptscriptstyle E} \quad \text{for all} \quad k \geq k_0 \text{ such that } \delta_k \leq d_1.$$

Thus, each iteration that satisfies these two conditions is very successful, and the trust-region radius may be expanded, i.e., $\delta_{k+1} \geq \delta_k$. As a consequence, it holds that, for all $j \geq 0$,

$$\delta_{k_0+j} \geq \min\{\gamma_{\scriptscriptstyle C} d_1, \delta_{k_0}\} := d_2. \tag{2.42}$$

The true reduction in the objective function is then bounded by

$$f(x_{k_0+j}) - f(x_{k_0+j+1}) \geq \frac{1}{2}\eta_A \tau_4 |\widehat{\lambda}| d_2^2 > 0$$

when iteration $k_0 + j$ is successful. If there are infinitely many successful iterations, then this contradicts Assumption 2.5.2. Thus, once $k$ is sufficiently large, all iterations fail to be successful, and $\delta_k$ is driven to zero by the algorithm. But this contradicts (2.42). the result follows. □

This theorem does not make any reference to the limit points of $\{x_k\}$. In fact, it does not even assume that limit points exist. Thus, the following assumption, which guarantees the existence of limit points, is made.

**Assumption 2.5.10.** *All iterates $\{x_k\}$ lie within a compact domain.*

This assumption is ubiquitous throughout optimization literature. It is often stated as an assumption on the objective function and starting point, in that Assumption 2.5.10 holds if the level set $\mathcal{L}(f(x))$ is compact, as Algorithm 2.2 guarantees that $f(x_{k+1}) \leq f(x_k)$ for all $k$

**Theorem 2.5.28.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, 2.5.7, 2.5.8, 2.5.9, and 2.5.10 hold. Then $\{x_k\}$ has at least one limit point $x^*$ at which the second-order necessary conditions hold.*

*Proof.* Theorem 2.5.27 ensures the existence of a subsequence of iterates $\{x_{k_i}\}$ such that $\{\lambda'_k(\nabla^2 f(x_{k_i}))\}$ converges to a nonnegative number. Under Assumption 2.5.10, $\{x_{k_i}\}$ must have a limit point $x^*$. Along with Assumption 2.5.6, this gives

$$\lambda'_{\mathcal{S}}(\nabla^2 f(x^*)) \geq 0 \quad \text{and} \quad \nabla f(x^*) = 0,$$

where the second inequality follows from Theorem 2.5.20. □

In order to make further claims, the behavior of Algorithm 2.2 when it yields a sequence $\{x_k\}$ that has a limit point $x^*$ that does not satisfy the second-order necessary conditions for optimality must be investigated.

**Lemma 2.5.29.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, 2.5.8, 2.5.7, and 2.5.9 hold. Suppose that $\{x_{k_i}\}$ is a subsequence of iterates that converges to a point $x^*$ at which the matrix*

$\nabla^2 f(x^*)$ *is indefinite, i.e.,* $\lambda'_{\mathcal{S}}(\nabla^2 f(x^*)) = \widehat{\lambda} < 0$. *Then*

$$\lim_{i \to \infty} \delta_{k_i} = 0$$

*and* $\lambda'_{k_i}(H_{k_i}) \leq \frac{1}{2}\widehat{\lambda}$ *for* $i$ *sufficiently large in every subsequence* $\{x_{k_i}\}$ *of iterates converging to* $x^*$.

*Proof.* First, consider that

$$\begin{aligned} 0 > \widehat{\lambda} &= \lambda'_{\mathcal{S}}(\nabla^2 f(x^*)) \\ &= \min_{||x||'_{\mathcal{S}}=1} x^{\mathrm{T}} \nabla^2 f(x^*) x \\ &= \min_{x \neq 0} \frac{x^{\mathrm{T}} \nabla^2 f(x^*) x}{(||x||'_{\mathcal{S}})^2} \\ &\geq \min_{x \neq 0} \frac{x^{\mathrm{T}} \nabla^2 f(x^*) x}{(||x||'_k)^2} = \lambda'_k(\nabla^2 f(x^*)) \end{aligned}$$

for all $k \in \mathcal{S}$, where the last inequality makes use of the fact that $||x||'_k \leq ||x||'_{\mathcal{S}}$ for all $k$, and therefore $-1/||x||'_k \leq -1/||x||'_{\mathcal{S}}$ for all $k$.

Now, Theorem 2.5.20 ensures that $\{||g_{k_i}||'_{k_i\star}\}$ converges to zero. Along with Assumption 2.5.8, this gives

$$\lim_{k \to 0} ||\nabla^2 f(x_{k_i}) - H_k||'_k = 0.$$

Assumption 2.5.1 then implies that $\lambda'_{k_i}(\nabla^2 f(x_{k_i}))$ is at most equal to $\widehat{\lambda}/2$ for $i$ sufficiently large. Using 2.5.8 again shows that for $i$ sufficiently large, $\lambda'_k(H_k) \leq \frac{1}{2}\widehat{\lambda}$.

In order to show that the trust-region radius converges to zero, first consider the case where $\mathcal{S}$ is finite. In this case, the result holds trivially, as the algorithm forces the radius to decrease. Next, assume that there are infinitely many successful iterations. To establish a contradiction, assume that there exists a subsequence $\{x_{k_i}\}$ converging to $x^*$ and a $\varepsilon \in (0, 1)$ such that $\delta_{k_i} \geq \varepsilon$ for all $i$. Without loss of generality, assume that the subsequence consists only of successful iterates. Notice that

$$\begin{aligned} -\lambda'_k(H_k)\delta_k &\geq \frac{1}{2}|\widehat{\lambda}|\delta_{k_i}^2 \\ &\geq \frac{1}{2}|\widehat{\lambda}|\varepsilon^2 := \delta. \end{aligned} \tag{2.43}$$

Assumption 2.5.9 then yields that, for $i$ sufficiently large,

$$q_{k_i}(x_{k_i}) - q_{k_i}(x_{k_i} + p_{k_i}) \geq \tau_4 \delta,$$

and therefore

$$f(x_{k_i}) - f(x_{k_i}) \geq \eta_A \tau_4 \delta.$$

This then implies that $f$ is unbounded below, which is a contradiction. Thus, the sequence of trust-region radii converges to zero, and the proof is complete. □

The main second-order convergence result can now be proved.

**Theorem 2.5.30.** *Suppose that assumptions 2.5.1, 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, 2.5.8, 2.5.7, and 2.5.9 hold. Let $x^*$ be any limit point of the sequence of iterates generated by Algorithm 2.2. Then $x^*$ satisfies the second-order necessary conditions for optimality.*

*Proof.* Theorem 2.5.20 ensures that $\lim_{k\to\infty} ||g_k||'_k = 0$. For the sake of establishing a contradiction, assume that

$$0 > \widehat{\lambda} = \lambda'_{\mathcal{S}}(\nabla^2 f(x^*)),$$

so that

$$\widehat{\lambda} \geq \lambda'_k(\nabla^2 f(x^*))$$

for all $k$. Assumption 2.5.8 then ensures that there exists a $k_0$ and a $\delta > 0$ such that

$$0 > \frac{1}{2}\widehat{\lambda} \geq \lambda'_k(H_k)$$

for each $k \geq k_0$ such that $||x_k - x^*||'_{\mathcal{S}} \leq \delta$. Let $\mathcal{K}$ be the subset of all such iteration indices. Assumption 2.5.9 then yields, for all $k \in \mathcal{K}$,

$$q_k(x_k) - q_k(x_k + p_k) \geq \frac{1}{2}\tau_4 |\widehat{\lambda}| \delta_k^2. \tag{2.44}$$

Consider now the ratio $\rho_k$. Lemma 2.5.24 applied to $\mathcal{K}$, along with the bound $||p_k||'_k \leq \kappa\delta_k$, ensures that there exists a $k_0$ and a $d_1$ such that

$$\rho_k \geq \eta_E \geq \eta_A \quad \text{for all } k \geq k_0 \text{ such that } ||x_k - x^*||'_{\mathcal{S}} \leq \delta \text{ and } \delta_k \leq d_1.$$

Thus, any such iteration is very successful. Now, let $\ell \geq k_0$ be an iteration such that $||x_\ell - x^*|| \leq \frac{1}{2}\delta$. Consider the sequence $\{x_{\ell+j}\}$. The two cases of the iterates remaining in $\{x : ||x - x^*||'_{\mathcal{S}} \leq \delta\}$ and leaving $\{x : ||x - x^*||'_{\mathcal{S}} \leq \delta\}$ are considered separately.

First, suppose that the iterates remain in $\{x : ||x - x^*||'_{\mathcal{S}} \leq \delta\}$. Then each iteration is very

successful if the trust-region radius is as small as $d_1$, and therefore

$$\delta_{\ell+j} \geq \min\{\gamma_C d_1, \delta_\ell\} := d_2 \tag{2.45}$$

for $j \geq 0$. Combining this with (2.44) yields

$$f(x_{\ell+j}) - f(x_{\ell+j+1}) \geq \frac{1}{2}\eta_A \tau_4 |\widehat{\lambda}| d_2^2 > 0 \tag{2.46}$$

whenever iteration $\ell + j$ is successful. If there are only finitely many successful iterations, then the trust-region radii converge to zero, contradicting the fact that $\delta_{\ell+j} \geq d_2 > 0$. Thus, there must be infinitely many successful iterations. In this case, however, (2.46) contradicts Assumption 2.5.2. Thus, the iterates must leave the ball $\{x : ||x - x^*||'_{\mathcal{S}} \leq \delta\}$.

If the sequence $\{x_{\ell+j}\}$ leaves the ball, then there must be a first successful iteration index $p \geq \ell$ such that $x_\ell = x_p \neq x_{p+1}$. Let $x_{q+1}$ be the first iterate to leave the ball. Then

$$\begin{aligned} \frac{\delta}{2} \leq ||x_{q+1} - x_{p+1}||'_{\mathcal{S}} &\leq \sum_{k=p, k\in\mathcal{S}}^{q} ||x_{k+1} - x_k||'_{\mathcal{S}} \\ &\leq c' \sum_{k=p, k\in\mathcal{S}}^{q} ||x_{k+1} - x_k||'_k \\ &\leq c'\kappa \sum_{k=p, k\in\mathcal{S}}^{q} \delta_k. \end{aligned} \tag{2.47}$$

Now, assume there exists a smallest integer $j$ such that $p \leq j \leq q$, and

$$\delta_j > \min\{\delta_{\max}, d_1\} := d_4. \tag{2.48}$$

If iteration $j$ is successful, (2.44) yields

$$f(x_p) - f(x_{q+1}) \geq f(x_j) - f(x_{j+1}) \geq \frac{1}{2}\eta_A \tau_4 |\widehat{\lambda}| d_4^2 := d_3 > 0.$$

If iteration $j$ is not successful, then $j > p$ (as $p$ is successful), and

$$d_4 < \delta_j \leq \gamma_E \delta_{j-1}.$$

Now, $j$ was assumed to be the first index such that (2.48) holds, so $\delta_{j-1} < \delta_j$. Thus, iteration $j - 1$ had

to have been successful. It then follows that

$$f(x_p) - f(x_{q+1}) \geq f(x_{j-1}) - f(x_j) \geq \frac{1}{2}\eta_A \tau_4 |\widehat{\lambda}| (d_4/\gamma_E)^2 := d_5 > 0 \tag{2.49}$$

On the other hand, if no such $j$ satisfying (2.48) exists, then all iterations between $p$ and $q$ are very successful, and

$$\sum_{k=p, k\in\mathcal{S}}^{q} \delta_k \leq \sum_{k=p, k\in\mathcal{S}}^{q} \frac{\delta_q}{\gamma_E^{q-k}} \leq \frac{\gamma_E}{\gamma_E - 1}\delta_q.$$

Combining this with (2.47) yields

$$\delta_q \geq \frac{(\gamma_E - 1)\delta}{2\gamma_E c' \kappa} := d_6.$$

Now, due to the fact that $q$ is the first iteration to leave the ball, it must have been successful. Therefore,

$$f(x_p) - f(x_{q+1}) \geq f(x_q) - f(x_{q+1}) \geq \eta_A \tau_4 |\widehat{\lambda}| d_6^2 := d_7 > 0.$$

Putting all bounds on $f(x_p) - f(x_{q+1})$ together yields

$$f(x_p) - f(x_{q+1}) > \min\{d_3, d_5, d_7\} > 0.$$

Assumption 2.5.2 then implies that the sequence $\{x_{\ell+j}\}$ may only leave the ball a finite number of times. Recall that it has already been shown that the iterates cannot remain in the ball, it must hold that

$$||x_k - x^*||'_{\mathcal{S}} \geq \frac{\delta}{2}$$

for all $k$ sufficiently large. However, $x^*$ is a limit point of the sequence $\{x_k\}$. This is a contradiction, and the result follows. □

This theorem implies that every limit point of the iterates produced by Algorithm 2.2 satisfies the second-order necessary conditions. Adding in Assumption 2.5.10 ensures that limit points do in fact exist, and thus the algorithm is guaranteed to converge to second-order stationary points.

The generality of Theorem 2.5.30 is applicable to a much larger class of methods than the standard trust-region convergence theorems using the standard steepest-descent direction. Consider, for instance, the case where $H_k = \nabla^2 f(x_k)$ for all iterations $k$, the norms $||\cdot||_k$ are taken to be $||\cdot||_{B_k}$ for some symmetric positive definite family of matrices $B_k$ such that $H_k$ and $B_k$ have the same sparsity pattern, and $||\cdot||'_k$ is taken to be $||\cdot||_{H_k+\mu_k B_k}$ for some bounded sequence of shifts $\mu_k$ such that each $H_k + \mu_k B_k$ is positive

definite. The steepest-descent direction induced by this family of norms, i.e., $g'_k = (H_k + \mu_k B_k)^{-1} g_k$ is now a much better approximation to the true solution than the direction $g_k$, and no modifications are needed for the results to prove convergence.

This section is concluded with a proof of convergence of the trust-region method when the trust-region subproblems are solved exactly, the matrix $H_k$ is chosen to be $\nabla^2 f(x_k)$ for all $k$, and the families of norms $\{||\cdot||_k\}$ and $\{||\cdot||'_k\}$ are both taken to be $\{||\cdot||_{B_k}$, where there exist positive constants $m$ and $M$ such that $mI \preceq B_k \preceq MI$ for all $k$. Let $d_k$ denote the solution to the subproblem

$$\begin{aligned} &\min_{d \in \mathbb{R}^n} Q_k(d) = \nabla f(x_k)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} \nabla^2 f(x_k) d \\ &\text{subject to} \quad ||d_k||_{B_k} \leq \delta_k. \end{aligned} \tag{2.50}$$

If $q_k$ is the quadratic model of $f$ at $x_k$, then

$$q_k(x_k) - q_k(x_k + d_k) > 0 \quad \text{and} \quad ||d_k||_{B_k} \leq \delta_k.$$

Theorem (4.1.1) implies the existence of $\sigma_k \geq 0$ such that

$$(\nabla^2 f(x_k) + \sigma_k B_k) d_k = -\nabla f(x_k) \quad \text{and} \quad \sigma_k(\delta_k - ||d_k||_{B_k}) = 0.$$

Therefore, $\sigma_k ||d_k||^2_{B_k} = \sigma_k \delta_k^2$, and

$$q_k(x_k) - q_k(x_k + d_k) = \frac{1}{2}(d_k^{\mathrm{T}}(\nabla^2 f(x_k) + \sigma_k B_k) d_k + \sigma_k \delta_k^2).$$

The sufficient decrease condition ensures that

$$f(x_k) - f(x_{k+1}) \geq \eta_A(q_k(x_k) - q_k(x_k + d_k)) = \frac{1}{2}\eta_A(d_k^{\mathrm{T}}(\nabla^2 f(x_k) + \sigma_k B_k) d_k + \sigma_k \delta_k^2) \tag{2.51}$$

**Theorem 2.5.31** ([16] Chapter 3)**.** *Let $f(x)$ be twice continuously differentiable in an open convex subset $\mathcal{D} \subseteq \mathbb{R}^n$, and assume that $x_0 \in \mathcal{D}$ is chosen so that the level set $\mathcal{L}(f(x_0))$ is compact. Let $\{x_k\}$ be the sequence defined by the trust-region method, where the sequence of trust-region matrices $\{B_k\}$ satisfies $mI \preceq B_k \preceq MI$, then either the algorithm terminates at $x_\ell \in \mathcal{L}(f(x_0))$ with $\nabla f(x_\ell) = 0$ and $\nabla^2 f(x_\ell) \succeq 0$, or $\{x_k\}$ has a limit point $x^*$ such that $x^* \in \mathcal{L}(f(x_0))$ with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$.*

*Proof.* If $\nabla f(x_\ell) = 0$ and $\nabla^2 f(x_\ell)$ is positive semidefinite for some iteration $\ell$, the algorithm terminates

with $d_\ell = 0$. Otherwise, $q_k(x_k) - q(x_k + d_k) > 0$ for all $k > 0$, thus $\{x_k\}$ is well-defined and lies in the set $\mathcal{L}(f(x_0))$. First, assume that a subsequence of $\{\sigma_k\}$ converges to zero. As $\mathcal{L}(f(x_0))$ is assumed to be compact, the same subsequence of $\{x_k\}$ must converge to some $x^* \in \mathcal{L}(f(x_0))$. The matrix $\nabla^2 f(x_k) + \sigma_k B_k$ is positive semidefinite, therefore $\{\nabla^2 f(x_k)\}$ is positive semidefinite matrix on the given subsequence. Now,

$$d_k^{\mathrm{T}}(\nabla^2 f(x_k) + \sigma_k B_k) d_k = (\nabla f(x_k))^{\mathrm{T}} (\nabla^2 f(x_k) + \sigma_k B_k)^{-1} \nabla f(x_k).$$

Let $U_k$ be a $B_k$-orthogonal matrix such that $U_k^{\mathrm{T}} \nabla f(x_k) U_k = \Lambda$, where $\Lambda$ is diagonal. Let $V_k = B_k U_k$, so that $\nabla f(x_k) = V_k \Lambda V_k^{\mathrm{T}}$. Then $\nabla f(x_k) = V_k U_k^{\mathrm{T}} \nabla f(x_k) = V_k y$, where $y = U_k^{\mathrm{T}} \nabla f(x_k)$. Thus, $||\nabla f(x_k)||_{B_k^{-1}} = ||y||$, and $V_k^{\mathrm{T}} (\nabla^2 f(x_k) + \sigma_k B_k)^{-1} V_k = \Lambda^{-1}$. Then,

$$\begin{aligned}
(\nabla f(x_k))^{\mathrm{T}} (\nabla^2 f(x_k) + \sigma_k B_k)^{-1} \nabla f(x_k) &= y^{\mathrm{T}} (\Lambda^{-1} + \sigma_k I) y \\
&\geq \frac{||y||^2}{||\Lambda|| + \sigma} \\
&= \frac{||\nabla f(x_k)||^2_{B_k^{-1}}}{||\nabla^2 f(x_k)||_{B_k} + \sigma_k}.
\end{aligned}$$

By (2.51), $d_k^{\mathrm{T}}(\nabla^2 f(x_k) + \sigma_k B_k) d_k$ and $\nabla f(x_k)$ converge to zero for the given subsequence.

It remains to show that $\sigma_k$ cannot remain bounded away from zero indefinitely. For the sake of contradiction, assume that $\sigma_k \geq \varepsilon$ for all $k$. By (4.1.1), this implies that $||d_k||_{B_k} = \delta_k$. Therefore,

$$q_k(x_k) - q_k(x_k + d_k) \geq \frac{1}{2} \sigma_k \delta_k \geq \frac{\varepsilon}{2} ||d_k||^2_{B_k}.$$

By the second-order mean value theorem,

$$|f(x_k + d_k) - q_k(x_k + d_k)| \leq \frac{1}{2} ||d_k||^2_{B_k} \max_{0 \leq t \leq 1} ||\nabla^2 f(x_k + d_k) - \nabla^2 f(x_k)||_{B_k}.$$

Recall that

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{q_k(x_k) - q_k(x_k + d_k)},$$

and that $q_k(x_k) = f(x_k)$. Combining these four equations gives

$$|\rho_k - 1| \leq \frac{1}{\varepsilon} \max_{0 \leq t \leq 1} ||\nabla^2 f(x_k + d_k) - \nabla^2 f(x_k)||_{B_k}. \tag{2.52}$$

The assumption that $\sigma_k \geq \varepsilon$ and (2.51) implies that $\{\delta_k\}$ converges to zero, and therefore $||d_k||_{B_k}$

converges to zero. By the assumptions on $B_k$, this implies that $||d_k|| \to 0$. As the parameter $\eta_A < 1$ in the trust-region algorithm, and $1/\varepsilon$ is bounded, (2.52) and the continuity of $\nabla^2 f(x)$ on the compact set $\mathcal{L}(f(x_0))$ imply that $\rho_k > \eta_A$ for $k$ sufficiently large. However, the updating rules for $\delta_k$ imply that $\delta_k$ stays bounded away from zero, as the step is accepted when $\rho_k \geq \eta_A$. This contradicts that $\delta_k \to 0$. Therefore $\sigma_k$ must not be bounded away from zero. □

Methods for solving the trust-region subproblem are explored in Chapter 4.

# Chapter 3

# Constrained Optimization

## 3.1 Introduction

In this section, problems of the form

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) \\ &\text{subject to } c_i(x) = 0 \quad \text{for all } i \in \mathcal{E}, \\ &\qquad \text{and } c_i(x) \geq 0 \quad \text{for all } i \in \mathcal{I}, \end{aligned} \tag{3.1}$$

are examined, where $\mathcal{E}$ and $\mathcal{I}$ are finite index sets of the equality and inequality constraints, respectively, as well as methods used to solve such problems. The list of methods is by no means exhaustive. Only methods that will help develop an understanding of the all-shifted penalty-barrier method are examined. Particular focus is placed on augmented Lagrangian methods for equality constraints and barrier methods for equality constraints. The only assumptions made on problem (3.1) is that $f$ and $c$ are both twice continuously differentiable. The feasible set of problem (3.1) is defined as

$$\Omega = \{x : c_i(x) = 0 \quad \text{for all } i \in \mathcal{E} \text{ and } c_i(x) \geq 0 \text{ for all } i \in \mathcal{I}\}.$$

Let $\mathcal{D} \subseteq \mathbb{R}^n$ denote the intersection of the domains of the functions $f(x)$, $c_1(x), \ldots, c_{m-1}(x)$, and $c_m(x)$, where $m$ is the number of constraints. For a given constraint $c_i(x)$, denote $\nabla c_i(x)$ as its gradient. Denote $c(x)$ as the vector valued function $c(x) = (c_1(x), \ldots, c_m(x))^{\mathrm{T}}$ and $J(x)$ as the Jacobian of the function $c$. The gradient of the objective function $f$ is denoted as $\nabla f(x)$, although the notation $\nabla f(x) = g(x)$ will often be more convenient. The Hessian of $f$ is similarly denoted as $\nabla^2 f(x)$. The symbol $H$ is reserved for the Hessian of the Lagrangian function of (3.1).

## 3.2 Equality Constraints

Consider the case where the index set of inequality constraints $\mathcal{I}$ is empty so that the problem being considered is

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) \\ &\text{subject to } c(x) = 0. \end{aligned} \tag{3.2}$$

As in the case of unconstrained problems, it is crucial to begin by defining the notion of what it means to be an optimal point.

**Definition 3.2.1** (Constrained Global Minimizer, [16] Chapter5)**.** A point $x^* \in \mathcal{D}$ is a *constrained global minimizer* of $f$ if $x^*$ is feasible, i.e., $x \in \Omega$, and $f(x^*) = \min\{f(x) \ : \ x \in \Omega\}$. If $x^*$ is a minimizer, then $f(x^*)$ is called the *global minimum* of $f$.

Unless $f$ is convex and $c$ is affine, the problem of finding a constrained global minimizer is generally intractable. Instead, most methods focus on finding a point that minimizes $f$ on a neighborhood of feasible points. Let $x^* \in \Omega$. Define $\mathcal{N}(x^*, \delta)$ to be the set of feasible points in an open $\delta$-ball about $x^*$, i.e., $\{x \in \mathcal{D} \ : \ ||x - x^*|| < \delta\}$. This motivates the following definition.

**Definition 3.2.2** (Constrained Local Minimizer, [16] Chapter 5)**.** Let $f$ be defined on $\mathcal{D} \subseteq \mathbb{R}^n$. Let $\mathcal{N}(x^*, \delta)$ denote the set $\mathcal{B}(x^*, \delta) \cap \Omega$, where $s^*$ and $\delta$ are such that $\mathcal{B}(x^*, \delta) \subseteq \mathcal{D}$. A point $x^*$ is a *constrained local minimizer* of $f$ if there exists a $\delta$ sufficiently small such that

$$f(x^*) \leq f(x) \text{ for all } x \in \mathcal{N}(x^*, \delta).$$

Such an $x^*$ is called *strict* if this inequality holds strictly except at $x^*$.

Note that if $x^*$ is locally unique, then it is necessarily a constrained local minimizer. As in the unconstrained case, it is important to consider points that are isolated minimizers, i.e., points that lie in a neighborhood in which there are no other minimizers of any kind. This is a stronger condition than simply being a strict minimizer.

**Definition 3.2.3** (Isolated Constrained Local Minimizer, [16] Chapter 5)**.** Let $\mathcal{N}(x^*, \delta) = \mathcal{B}(x^*, \delta) \cap \Omega$. A constrained local minimizer $x^*$ is *isolated* if there is a $\delta$ sufficiently small such that $x^*$ is the unique constrained local minimizer in $\mathcal{N}(x^*, \delta)$.

Open balls are not the only neighborhoods that can be considered.

**Definition 3.2.4** (Constrained Local Minimizer (2), [16] Chapter 5)**.** The point $x^*$ is a *constrained local minimizer* of $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ if there exists a compact set $\mathcal{S}$ such that

$$x^* \in \text{int}(S) \cap \Omega \text{ and } f(x^*) = \min\{f(x) \ : \ x \in S \cap \Omega\}.$$

### 3.2.1 Optimality Conditions

As in the case of unconstrained optimization, the above definitions of solutions are not particularly useful for developing methods, as certifying optimality would require checking all points in a neighborhood of a potential solution. The key observation is that if a point $x^*$ is to be locally optimal, then the objective function cannot decrease on any path that remains in the feasible set. Such a path is called a *feasible path.*

**Definition 3.2.5** (Feasible Path, [16] Chapter 5)**.** A *feasible path* is a directed, twice-differentiable curve $x(\alpha)$ starting at a point $x^*$, parameterized by the scalar $\alpha$ such that

1. $x(0) = x^*$ and $c(x(\alpha)) = 0$ for all $\alpha$ satisfying $0 \leq \alpha < \widehat{\alpha}$, where $\widehat{\alpha} > 0$, and
2. The curve is regular, i.e., the tangent $p$ to the curve $x(\alpha)$ is never equal to zero.

If $x(\alpha)$ is to remain feasible as $\alpha$ increases, each constraint function must remain equal to zero for all $\alpha \in [0, \widehat{\alpha})$. Thus, the derivative of the function $c(x(\alpha))$ must be zero for all $\alpha \in [0, \widehat{\alpha})$. Taking the derivative of the $i$-th constraint along $x(\alpha)$ gives

$$\frac{d}{d\alpha} c_i(x(\alpha))\bigg|_{\alpha=0} = \nabla c_i(x)^{\mathrm{T}} p = 0,$$

where $p$ is the tangent vector of $x(\alpha)$ at $\alpha = 0$, i.e.,

$$p = \lim_{\alpha \to 0_+} \frac{1}{\alpha}(x(\alpha) - x(0)).$$

If this holds for each constraint $i$, then

$$J(x^*)p = 0. \tag{3.3}$$

Thus, it holds that the tangent $p$ of a feasible path lies in the null space of the Jacobian matrix $J(x)$. A similar result can be established for *feasible sequences.*

**Definition 3.2.6** (Feasible sequences, [16] Chapter 5)**.** Let $x \in \Omega$. A *feasible sequence* converging to $x$ is a sequence $\{x_k\}_{k=0}^{\infty}$ such that for all $k$, $x_k$ is feasible and $x_k \neq x^*$.

Consider the related sequence $\{p_k\}$, where $p_k$ is given by $x_k = x + t_k p_k$, with $t_k$ a normalizing constant so that $||p_k|| = 1$. Then $\lim_{k\to\infty} t_k = \lim_{k\to\infty} ||x - x_k|| = 0$. The sequence $\{p_k\}$ then lies in a compact set, and therefore it holds that a subsequence converges to some vector $p$. Without loss of generality, it suffices to only consider this subsequence. A nonzero vector $p$ is a tangent of $\Omega$ at $x \in \Omega$ if there exists a feasible sequence $\{x_k\}$ such that $x_k \neq x$ for all $k$ and

$$\lim_{k\to\infty} \frac{1}{||x_k - x||}(x_k - x) = \frac{1}{||p||}p.$$

In order for each vector in the feasible sequence to be feasible, it must hold that each constraint satisfies $c_i(x_k) = 0$ for all $k$. Therefore,

$$c_i(x_k) = c_i(x + t_k p_k) = 0.$$

Taking the limit as $k$ goes to infinity and recalling that both $c(x^*)$ and $c(x^* + t_k p_k)$ are zero gives

$$\lim_{k\to\infty} \frac{c_i(x^* + t_k p_k) - c(x^*)}{t_k} = \nabla c_i(x^*)^{\mathrm{T}} p = 0,$$

where $p = \lim_{k\to\infty} p_k$. As this must hold for each constraint, it again holds that $J(x^*)p = 0$, (see (3.3)).

The identity (3.3) show that the vector $p$ lies in the null space of the matrix $J(x)$. Let $Z(x)$ denote a matrix whose columns form a basis for the null space of $J(x)$ so that $J(x)Z(x) = 0$. The vector $p$ can then be written as a linear combination of the columns of $Z(x)$, i.e., $p = Z(x)p_z$ for some $z$.

Of particular interest is the collection of all tangent vectors corresponding to all feasible sequences at $x$. This leads to the notion of the *tangent cone* at a feasible point $x$.

**Definition 3.2.7** (Tangent Cone, [16] Chapter 5)**.** Let $\mathcal{T}^+(x)$ denote the set of all tangent vectors associated with a feasible sequence starting at a feasible point $x$. Then $\mathcal{T}(x) = \mathcal{T}^+(x) \cup \{0\}$ is known as the *tangent cone* of $c$ at $x$.

Readers familiar with the definition of the *tangent space* of a submanifold of $\mathbb{R}^n$ defined by $\{x \in \mathbb{R}^n : c(x) = 0\}$ for some sufficiently regular function $c : \mathbb{R}^n \to \mathbb{R}^m$ may wonder why the terminology used here is tangent cone as opposed to tangent space. The reason for this is that no assumptions about $c(x)$ are made beyond the assumption that it is twice continuously differentiable. Thus it can only be concluded that $\mathcal{T}(x)$ is a cone and not necessarily a vector space. The tangent cone provides the tools needed to define usable conditions of optimality.

**Theorem 3.2.1** ([16] Chapter 5)**.** *Assume that $f$ and $c$ are differentiable. Let $x^* \in \Omega$ be a feasible point*

*at which at least one feasible sequence exists.* $x^*$ *is a local solution to* (3.2) *only if* $\nabla f(x^*)^{\mathrm{T}} p \geq 0$ *for every* $p \in \mathcal{T}(x^*)$.

*Proof.* Let $\{x_k\}$ be a feasible sequence converging to $x^*$. The definition of a constrained local minimizer states that $f(x_k) \geq f(x^*)$ for all $k$ sufficiently large. Consider the sequences $\{t_k\}$ and $\{p_k\}$ given by $x_k = x^* + t_k p_k$, where $t_k$ are normalizing constants such that $||p_k|| = 1$. Then

$$f(x^* + t_k p_k) \geq f(x^*)$$

for all sufficiently large $k$. By assumption, $f$ is differentiable at $x^*$, so

$$\lim_{k \to 0} \frac{f(x^* + t_k p_k) - f(x^*)}{t_k} = \nabla f(x^*)^{\mathrm{T}} p, \text{ with } p = \lim_{k \to \infty} p_k.$$

It follows from the two above relations that $\nabla f(x^*)^{\mathrm{T}} p \geq 0$. □

Thus, it holds that $x$ is optimal only if $\nabla f(x^*)^{\mathrm{T}} p \geq 0$ for each $p \in \mathcal{T}(x)$. Consider the sets

$$\mathcal{T}^{\star}(x) = \{g \ : \ g^{\mathrm{T}} p \geq 0 \text{ for all } p \in \mathcal{T}(x)\}.$$

This set is referred to as the *dual cone* of $\mathcal{T}(x)$. It is clear from the definition of the dual cone that $x$ is optimal only if $\nabla f(x) \in \mathcal{T}^{\star}(x)$. However, this notion of optimality is still not particularly useful. What is needed is a way to certify that $\nabla f(x)$ lies within the dual cone without having to actually check every vector in the dual cone.

From (3.3), it is clear that $T(x) \subseteq \text{null}(J(x))$. Null$(J(x))$ is itself a tangent cone (actually a tangent space) of the linearized constraint at $x$. Consider the Taylor series of constraint function $c(x)$ at a point $x_0$:

$$c(x) = c(x_0) + J(x_0)(x - x_0) + \mathcal{O}(||x - x_0||_2).$$

Ignoring the higher order terms, the linearized constraint function at $x_0$ is defined as

$$c_{L}(x; x_0) = c(x_0) + J(x_0)(x - x_0). \tag{3.4}$$

If $c$ is nonlinear with $c(x_0) \neq 0$, then $c_{L}(x; x_0) = 0$ defines an affine approximation of $c(x) = 0$ at $x_0$. If $x_0$ is feasible, and $c(x_0) = 0$, then $c_{L}(x; x_0) = 0$ defines a set consisting of the intersection of $m$ hyperplanes.

If $x_0$ is clear from context, this linearized constraint is written as $c_{L}(x)$. A key property of the

linearized constraint is that every vector $p \in \text{null}(J(x))$ is tangent to a feasible sequence (with respect to the linearized constraint) at a point $x = \widehat{x}$. It is fairly obvious that the tangent cone for a linearized constraint is equivalent to the null space of $J(x)$.

**Lemma 3.2.2** ([16] Chapter 5)**.** *If $\mathcal{T}_L(x)$ denotes the tangent cone of the linearized constraint $c_L(x) = 0$, then $\mathcal{T}_L(x) = \text{null}(J(x)) = \{p \ : \ p = Z(x)p_z\}$, where the columns of $Z(x)$ form a basis for the null space of $J(x)$.*

Lemma 3.2.2 implies then that $\mathcal{T}(x) \subseteq \mathcal{T}_L(x)$. The linearized tangent cone gives the machinery needed to define a usable notion of optimality.

**Theorem 3.2.3** (Lagrange Multipliers for problem (3.2), [16] Chapter 5)**.** *Assume that $f$ and $c$ are differentiable at a feasible point $x^*$. Then*

$$\nabla f(x^*)^{\mathrm{T}} p \geq 0 \text{ for all } p \in \text{null}(J(x^*))$$

*if and only if there exists a vector $y^*$, called the Lagrange multipliers, such that*

$$\nabla f(x^*) - J(x^*)^{\mathrm{T}} y = 0.$$

*Proof.* If $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$ for some $y^*$, then for all $p \in \text{null}(J(x^*))$, it holds that

$$\nabla f(x^*)^{\mathrm{T}} p = (y^*)^{\mathrm{T}} J(x^*) p = 0.$$

On the other hand, suppose there is no $y^*$ such that $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$. Then $\nabla f(x^*) \notin \text{range}(J(x^*)^{\mathrm{T}})$, so $\nabla f(x^*)$ can be uniquely decomposed into

$$\nabla f(x^*) = g_R + g_N, \quad \text{where} \quad g_N \neq 0,$$

and $g_N \in \text{null}(J(x^*))$ and $g_R \in \text{range}(J(x^*)^{\mathrm{T}})$. Choosing $p = -g_N$ satisfies $J(x^*)p = 0$. Then

$$\nabla f(x^*)^{\mathrm{T}} p = -||g_N||_2^2 < 0.$$

The result follows. □

The Lagrange multipliers $y^*$ can be shown to be unique if $J(x^*)$ has full row rank. The existence of Lagrange multipliers provides a computable certificate of optimality.

**Definition 3.2.8** (KKT point for problem (3.2), [16] Chapter 5)**.** Let $x^* \in \Omega$ such that there exists a vector $y^* \in \mathbb{R}^m$ such that $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$. The point $x^*$ is referred to as a *first-order KKT point* for (3.2).

It is important to note that being a KKT point is not always a necessary condition for optimality. The definition of KKT points depended on the linearized tangent cone and the true tangent cone at a point $x^*$ to have equivalent dual cones, i.e., $\mathcal{T}^\star(x^*) = \mathcal{T}_L^\star(x^*)$. For most problems, this is the case. However, problems can be constructed for which this does not hold. In practice, it is useful to assume that the constraint function $c$ possesses a regularity condition that is verifiable in practice, and implies that a problem is not one of these pathological cases. Such a condition is called a *constraint qualification*. Optimality conditions are often phrased using the following language: if $x^*$ is a local minimizer, then $x^*$ is a KKT point or a constraint qualification does not hold. There are many different constraint qualifications that are utilized when formulating algorithms. Different constraint qualifications tend to trade strictness for practical verifiability, i.e., a constraint qualification may only apply to a small subset of problems for which a local minimizer is a KKT point, but the qualification is easy to verify in practice. Conversely, a constraint qualification may apply to a vast number of problems but may be quite difficult to verify. Only a few commonly used constraint qualifications are listed here.

**Definition 3.2.9** (Guignard constraint qualification, [16] Chapter 5)**.** The Guignard constraint qualification holds at $x$ if $x \in \Omega$ and $\mathcal{T}^\star(x) = \mathcal{T}_L^\star(x)$.

Working with the dual cones is not as convenient as working with the original cones. This motivates the following definition.

**Definition 3.2.10** (Abadie Constraint Qualification, [16] Chapter 5)**.** The Abadie constraint qualification holds at $x$ if $x \in \Omega$ and $\mathcal{T}(x) = \mathcal{T}_L(x)$.

The Abadie constraint qualification implies the Guignard constraint qualification, but not vice versa. However, even the Abadie constraint qualification is difficult to verify in practice. A much more convenient constraint qualification is the following:

**Definition 3.2.11** (Linearly independent constraint qualification (LICQ), [16] Chapter 5)**.** Assume that the constraint function $c(x) : \mathbb{R}^n \to \mathbb{R}^m$ is continuously differentiable. A sufficient condition for the constraint qualification to hold at a feasible point $x$ is that the rows of $J(x^*)$ are linearly independent.

Constraint qualifications and the KKT conditions are enough to establish first-order necessary conditions for optimality, as the following result shows.

**Theorem 3.2.4** (First-Order Necessary Conditions for (3.2), [16] Chapter 5)**.** *If a constraint qualification holds at $x^*$, then $x^*$ is a local minimizer of* (3.2) *only if $x^*$ is a KKT point.*

Note that the KKT conditions consists of $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$ and the fact that $x^* \in \Omega$, i.e., $c(x^*) = 0$. These two conditions together make up the KKT conditions. Both conditions can be described in terms of a single function called the Lagrangian function

$$L(x, y) = f(x) - y^{\mathrm{T}} c(x).$$

The KKT conditions then become a stationarity condition of $L$, i.e.,

$$\nabla L(x, y) = \begin{pmatrix} \nabla_x L(x, y) \\ \nabla_y L(x, y) \end{pmatrix} = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ -c(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

In general, a solution $x^*$ and an associated Lagrange multiplier $y^*$ that satisfies the KKT conditions do not constitute a minimizer of $L$. Thus, algorithms for solving (3.2) cannot simply attempt to minimize the Lagrangian. Instead, algorithms typically attempt to find critical points of $L$ that satisfy both first and second-order optimality conditions.

**Second-Order Optimality Conditions**

In the unconstrained case, second-order optimality conditions focused on properties of the second derivative of the objective function at points that satisfied first-order optimality conditions. The same is true here, except the second derivatives of the constraint function $c$ must be considered alongside the second derivatives of the objective function $f$. The second derivatives of the Lagrangian play a crucial role.

At a point $(x, y)$, the Hessian of the Lagrangian with respect to the $x$ variables is given by

$$H(x, y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 c_i(x). \tag{3.5}$$

This matrix plays a critical role in the following result.

**Theorem 3.2.5** (Second-Order Necessary Conditions for (3.2), [16] Chapter 5)**.** *If a constraint qualification holds at $x^*$, then $x^*$ is a local solution of* (3.2) *if*

1. $c(x^*) = 0$,
2. *there exist a vector $y^*$ such that $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$, and*

3. $p^{\mathrm{T}} H(x^*, y^*)p \geq 0$ *for every* $p \in \mathrm{null}(J(x^*))$.

*Proof.* The first two conditions are simply the first-order necessary conditions. Let $p$ satisfy $J(x^*)p = 0$. If there is no such $p$, the result holds. Let $x^*(\alpha)$ be a twice continuous differentiable path such that $x(0) = x^*$ and $x'(0) = p$. Such a path is guaranteed to exist by the constraint qualification. Let $v = x''(0)$. Due to the fact that $c(x(\alpha)) = 0$, it holds that

$$\frac{d^2}{d\alpha^2} c_i(x(\alpha))\bigg|_{\alpha=0} = \nabla c_i(x^*)^{\mathrm{T}} v + p^{\mathrm{T}} \nabla^2 c_i(x^*)p. \tag{3.6}$$

By the second condition, $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$ for some $y^*$. Then

$$\frac{d}{d\alpha} f(x(\alpha))\bigg|_{\alpha=0} = \nabla f(x^*)^{\mathrm{T}} p = (y^*)^{\mathrm{T}} J(x^*)p = 0. \tag{3.7}$$

Therefore, $x^*$ is a stationary point along the feasible path. In order for $x^*$ to be a local minimizer along this path, then the second order necessary condition for an unconstrained minimizer must hold, i.e.,

$$\frac{d^2}{d\alpha^2} f(x(\alpha))\bigg|_{\alpha=0} \geq 0.$$

Using (3.7) and the definition of $v$, it holds that

$$\frac{d^2}{d\alpha^2} f(x(\alpha))\bigg|_{\alpha=0} = (y^*)^{\mathrm{T}} J(x^*)v + p^{\mathrm{T}} \nabla^2 (x^*)p \geq 0. \tag{3.8}$$

Combining (3.6) and (3.8) yields

$$\frac{d^2}{d\alpha^2} f(x(\alpha))\bigg|_{\alpha=0} = -p^{\mathrm{T}} \left( \sum_{i=1}^{m} y_i^* \nabla^2 c_i(x^*) \right) p + p^{\mathrm{T}} \nabla^2 f(x^*)p = p^{\mathrm{T}} H(x, y)p \geq 0.$$

□

A point satisfying the conditions of Theorem 3.2.5 is called a *second order KKT point* of (3.2). The following result gives sufficient conditions for a point $x^*$ to be a local minimizer of (3.2).

**Theorem 3.2.6** (Second-Order Sufficient Conditions for (3.2), [16] Chapter 5)**.** *A point* $x^*$ *is a strict local minimizer of* (3.2) *if*

1. $x^*$ *is feasible, i.e.,* $c(x^*) = 0$,

2. *there exists a vector* $y^*$ *such that* $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$, *and*

*3. the strict inequality $p^{\mathrm{T}} H(x^*, y^*)p > 0$ holds for every $p \neq 0$ such that $J(x^*)p = 0$.*

*Proof.* Consider a feasible sequence $\{x_k\}$ converging to $x^*$. Define the sequence $\{t_k\}$ and $\{p_k\}$ via the equation $x_k = x^* + t_k p_k$, where $t_k$ is a normalizing constant so that $||p_k|| = 1$. It follows that $t_k \to 0$. The sequence $\{p_k\}$ is bounded, and therefore has a convergent subsequence. In what follows, $\{p_k\}$ will denote the subsequence converging to $p$. The Taylor-series expansion with the integral form of the remainder of the constraint function $c$ at $x^*$ gives

$$c(x_k) = c(x^*) + t_k J(x_k)p_k + t_k \int_0^1 \big(J(x^* + \xi t_k p_k) - J(x^*)\big)p_k \, d\xi.$$

By construction, $c(x_k) = c(x^*)$, so

$$J(x^*)p_k = -\int_0^1 \big(J(x^* + \xi t_k p_k) - J(x^*)\big)p_k \, d\xi.$$

Taking the limit as $k \to \infty$ gives $J(x^*)p = 0$.

Suppose now that $x^*$ is not a strict minimizer. Then there exists a feasible sequence $\{x_k\}$ such that $x_k \to x^*$ and $f(x^*) \geq f(x_k)$ for $k$ sufficiently large. Again using a Taylor-series with an integral remainder gives

$$0 \geq f(x_k) - f(x^*) = t_k \nabla f(x_k)^{\mathrm{T}} p_k + t_k^2 \int_0^1 p_k^{\mathrm{T}} \nabla^2 (x^* + \xi t_k p_k) p_k (1 - \xi) \, d\xi, \tag{3.9}$$

and

$$0 = t_k \nabla c_i(x^*)^{\mathrm{T}} p_k + t_k^2 \int_0^1 p_k^{\mathrm{T}} \nabla^2 c_i (x^* + \xi t_k p_k) p_k (1 - \xi) \, d\xi.$$

Multiplying the above expression by $y_i^\star$ for each index $i$ and subtracting the result from (3.9), and using $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$ gives

$$0 \leq t_k^2 \int_0^1 p_k^{\mathrm{T}} H(x^* + \xi t_k p_k, y^*) p_k (1 - \xi) \, d\xi.$$

Taking limits as $k \to \infty$ yields $p^{\mathrm{T}} H(x^*, y^*)p \leq 0$. Therefore, in order for $x^*$ to be a strict minimizer, it must hold that $p^{\mathrm{T}} H(x^*, y^*)p > 0$. The result follows. □

It is important to note that the above result does not require that a constraint qualification hold at $x^*$. If a constraint qualification is included, as stronger statement about $x^*$ can be made, as seen in the next result.

**Theorem 3.2.7** (Second-Order Sufficient Conditions for an Isolated Minimizer of (3.2), [16] Chapter 5)**.** *A point $x^*$ is an isolated local minimizer of* (3.2) *if*

1. *$x^*$ is feasible, i.e., $c(x^*) = 0$,*
2. *there exists a vector $y^*$ such that $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$,*
3. *the strict inequality $p^{\mathrm{T}} H(x^*, y^*)p > 0$ holds for every $p \neq 0$ such that $J(x^*)p = 0$, and*
4. *the constraint gradients at $x^*$ are linearly independent.*

*Proof.* The first three conditions are the sufficient conditions of a strict local minimizer. All that remains to show is that $x^*$ is isolated. Conditions (1) and (2) state that the point $(x^*, y^*)$ is a zero of the function $F(x, y) = \nabla L(x, y)$. The result follows from applying the implicit function theorem to $F$ and combining it with condition (3). □

### 3.2.2 Augmented Lagrangian Methods

In this section, augmented Lagrangian methods are briefly reviewed. Techniques from these methods will be applied to the equality constraints in the all-shifted primal-dual penalty-barrier trust-region method.

Augmented Lagrangian methods can be derived using a variety of tools. They can be viewed as a penalty method applied to a version of (3.2) with the constraints shifted by some optimal shift. If a problem is convex, it can be viewed as applying a proximal method to the corresponding concave dual problem. They can also be viewed as applying unconstrained optimization techniques to a sequence of modified Lagrangian functions with additional terms added to ensure positive curvature in the necessary directions.

Assume that the sufficient optimality conditions of Theorem 3.2.6 hold at a point $x^*$. As previously mentioned, $x^*$ is a stationary point of $L(x, y^*)$, where $y^*$ is the Lagrange multiplier, but not necessarily a minimizer. By (3.2.6), the Hessian of $L(x, y)$ at $(x^*, y^*)$ is positive definite when restricted to the subspace null$(J(x^*))$. Therefore, the Lagrangian can only have directions of negative curvature in complementary space range$(J(x^*))$. This suggests that the Lagrangian can be augmented by an additional term to correct for this negative curvature. Although there are many possible ways to augment the Lagrangian, the most commonly seen approach is to add a *quadratic penalty* term, yielding

$$L_A(x; y^E, \rho) = L(x, y^E) + \frac{\rho}{2}||c(x)||_2^2, \tag{3.10}$$

for some parameter $\rho$, referred to as the *penalty parameter*. This penalty term has the effect of increasing the potentially negative eigenvalues of $H(x, y)$ but leaving eigenvalues that are guaranteed to be positive unchanged.

**Theorem 3.2.8** ([16] Chapter 5)**.** *Assume that $x^*$ satisfies second-order sufficient conditions for a strict local minimizer of* (3.2)*. Let $y^*$ be the Lagrange multipliers at $x^*$. There is a finite $\bar{\rho}$ such that, for each $\rho > \bar{\rho}$, a solution $x^*$ of* (3.2) *is an isolated unconstrained local minimizer of the augmented Lagrangian function $L_A(x; y^*, \rho)$.*

*Proof.* The gradient of $L_A(x; y, \rho)$ is given by

$$\nabla L_A(x; y, \rho) = \nabla f(x) - J(x)^{\mathrm{T}}(y - \rho c(x)),$$

and the Hessian is given by

$$\nabla^2 L_A(x; y, \rho) = H(x, y - \rho c(x)) + \rho J(x)^{\mathrm{T}} J(x).$$

At a local minimizer $x^*$ of (3.2), $c(x^*) = 0$, and therefore

$$\nabla L_A(x^*; y^*, \rho) = \nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0,$$

and

$$\nabla^2 L_A(x^*; y^*, \rho) = H(x^*, y^*) + \rho J(x^*)^{\mathrm{T}} J(x^*).$$

The second order sufficient conditions imply that if $Z$ is a matrix whose columns form a basis of $\mathrm{null}(J(x^*))$, then $Z^{\mathrm{T}} H(x^*, y^*) Z \succ 0$. Therefore, there exists a constant $\bar{\rho}$ such that $H(x^*, y^*) + \rho J(x^*)^{\mathrm{T}} J(x^*)$ is positive definite for all $\rho > \bar{\rho}$. Therefore, $x^*$ is an isolated minimizer of $L_A(x; y^*, \rho)$. □

The above result is one of the main reasons augmented Lagrangian methods are preferred over more classical penalty methods. Furthermore, note that the above result does not require that the constraint gradients are linearly independent at $x^*$.

One drawback of Theorem 3.2.8 is that $y^*$ is not known in advance. Therefore, practical augmented Lagrangian methods make do with an estimate of the Lagrange multiplier $y^E$ that is updated along with the penalty parameter. This forms the general outline of augmented Lagrangian methods:

1. Solve $\min_{x \in \mathbb{R}^n} L_A(x; y^E, \rho)$.

2. Check for optimality.

3. Update $\rho$ and $y^E$.

These simple steps are iterated until some approximate optimality conditions are satisfied. A simple form of the update for $y^E$ can be seen in the first-order optimality conditions of $L_A(x; y^E, \rho)$. Recall that

$$\nabla L_A(x_j; y^E, \rho) = \nabla f(x_j) - J(x)^{\mathrm{T}}(y^E - \rho c(x_j)).$$

Clearly, the vector $y^E_{j+1} = y^E_j - \rho c(x_j)$ satisfies $\nabla f(x_j) - J(x_j)^{\mathrm{T}} y^E_{j+1} = 0$. Thus, this update can be used as a suitable approximation of $y^*$. This expression is known as the *first-order* multipliers, due to the fact that

$$||y^E_{j+1} - y^*|| \leq M||y^E_j - y^*||$$

for $j$ sufficiently large. The augmented Lagrangian method using first-order multipliers can only converge as fast as $y^E_{j+1}$ converges to $y^*$. Thus, even a quadratically convergent technique will be slowed down by this update rule. Although there exists other multiplier estimates that avoid this problem at the cost of additional computation, they are not discussed here.

Note that the augmented Lagrangian method treats the primal variables $x$ and the dual variables (Lagrange multipliers) $y$ separately. Alternative approaches operate by treating both sets of variables as variables minimized that minimize an objective function. One such approach is the *primal-dual augmented-Lagrangian method*.

## The Primal-Dual Augmented-Lagrangian Method

In [15], Gill et al. propose methods for solving problem (3.2) centered around the *primal dual augmented Lagrangian*

$$M(x, y; y^E, \mu^P) = f(x) - c(x)^{\mathrm{T}} y^E + \frac{1}{2\mu^P}||c(x)||_2^2 + \frac{1}{2\mu^P}||c(x) + \mu^P(y - y^E)||_2^2. \tag{3.11}$$

The penalty parameter $\rho$ in this equation has been replaced with $1/\mu^P$. $\mu^P$ will be referred to as the penalty parameter from now on. This function $M$ will be incorporated into the merit function used in the all-shifted primal-dual penalty-barrier merit function. The general framework of a method utilizing $M$ would be as follows:

1. Take a step $(x_{k+1}, y_{k+1}) = (x_k + p_k, y_k + q_k)$ towards a solution of the unconstrained problem $\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} M(x, y; y^E_k, \mu^P_k)$

2. Check for optimality.

3. Update $y^E$ and $\mu^B$.

In this case, the update to $y^E$ is based on the most recently computed $y_k$, as opposed to some estimate based on $x_j$. Although the details of the overall method are not investigated here (see [15] for more details), some properties of $M$ are examined. Consider the first and second-order derivatives of $M$ in both $x$ and $y$:

$$\nabla M = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}}(2(y^E - \frac{1}{\mu^P}c(x)) - y) \\ \mu^P(y - (y^E - \frac{1}{\mu^P}c(x))) \end{pmatrix}.$$

Let $\pi^y = (y^E - \frac{1}{\mu^P}c(x))$. Note that $\pi^y$ is equivalent to the first-order multiplier update detailed in the augmented Lagrangian method. Then

$$\nabla M = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}}(2\pi^y - y) \\ \mu^P(y - \pi^y) \end{pmatrix},$$

and

$$\nabla^2 M = \begin{pmatrix} H(x, 2\pi^y - y) + \frac{2}{\mu^P}J(x)^{\mathrm{T}}J(x) & J(x)^{\mathrm{T}} \\ J(x) & \mu^P I \end{pmatrix}.$$

From the gradient of $M$, it can be observed that at a local minimizer of $M$, $\pi^y - y = 0$. After some rearrangement, this becomes the perturbed feasibility condition $c(x) = \mu^P(y^E - y)$. Thus, it can be shown that minimizing a sequence of primal-dual augmented Lagrangian functions is equivalent to solving a sequence of perturbed first-order KKT conditions

$$\begin{aligned} \nabla f(x) - J(x)y &= 0, \\ c(x) &= \mu^P(y^E - y). \end{aligned}$$

This basic structure will be expanded upon once methods for managing inequality constraints have been discussed.

## 3.3 Inequality Constraints

Now, consider the case where the set of equality constraints $\mathcal{E}$ is empty, so that the problem under consideration becomes

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) \\ &\text{subject to } c(x) \geq 0. \end{aligned} \tag{3.12}$$

As in the unconstrained and equality-constrained case, the first task is to characterize the notion of optimality for problem (3.12).

### 3.3.1 Optimality Conditions

**Definition 3.3.1** ([16] Chapter 6)**.** A constraint $c_i(x) \geq 0$ is said to be *active* at a point $x^*$ if $c_i(x^*) = 0$, and *inactive* if $c_i(x^*) > 0$.

As before, let $\Omega$ denote the feasible set $\{x \ : \ c_i(x) \geq 0 \text{ for all } i \in \mathcal{I}\}$. The feasible set is said to have a *strict interior* if there exists a point $\bar{x} \in \Omega$ such that all constraints are inactive at $\bar{x}$. The discussion of optimality conditions proceeds similar to the equality constrained case, with the distinction that feasible paths and sequences are not bound to lie on the constraint surface for all constraints.

**Definition 3.3.2** ([16] Chapter 6)**.** Let $x \in \Omega$ be a feasible point. The set $\mathcal{A}(x) = \{i \in \mathcal{I} : c_i(x) = 0\}$ is referred to as the *active set* at $x$, and consists of the index set of the active constraints at a point $x$.

The notation $c_a(x)$ is used to denote the subset of constraints in the active set evaluated at $x$. Similarly, $J_a(x)$ is used to denote the gradient of the active constraints at $x$. Let $m_a(x) = |\mathcal{A}(x)|$.

**Definition 3.3.3** (Feasible path, [16] Chapter 6)**.** Let $x \in \Omega$. A feasible path is a twice-differentiable curve $x(\alpha)$ such that

1. $x(0) = x$ and $c(x(\alpha)) \geq 0$ for all $\alpha$ satisfying $0 \leq \alpha < \widehat{\alpha}$ for some $\widehat{\alpha} > 0$, and
2. the tangent vector $p = x'(0)$ to the feasible path at $x$ is nonzero.

**Definition 3.3.4** (Feasible sequences, [16] Chapter 6)**.** Let $x \in \Omega$. A feasible sequence converging to $x$ is a sequence $\{x_k\}$ such that, for all $k$, $c(x_k) \geq 0$, $x_k \neq x$, and $\lim_{k\to\infty} x_k = x$.

The tangent of a feasible sequence is defined by constructing sequence $\{t_k\}$ and $\{p_k\}$ via $x_k = x + t_k p_k$, where $t_k$ is a normalizing constant so that $||p_k|| = 1$. The sequence $\{p_k\}$ is therefore bounded, so there exists a convergent subsequence. Without loss of generality, the convergent subsequence converges to a vector $p$, the tangent vector of $\{x_k\}$.

**Definition 3.3.5** (The tangent cone, [16] Chapter 6)**.** Let $\mathcal{T}^+(x)$ denote the set of tangent vectors associated with every feasible sequence converging to a feasible point $x$. The tangent cone is then $\mathcal{T}(x) = \mathcal{T}^+(x) \cup \{0\}$.

If a constraint $c_i(x)$ is inactive at $x$, then there exists a neighborhood of $x$ in which $c_i(x)$ is inactive for all points in this neighborhood, thus it does not place any restrictions on tangent vectors. On the other hand, active constraints limit the directions of potential tangent vectors of feasible sequences. A feasible path $x(\alpha)$, or a feasible sequence $\{x_k\}$, is called *binding* if some constraint $c_i(x)$ remains feasible for all points along the path or sequence. If $p$ is tangent to a binding feasible sequence, then that constraint functions as an equality constraint for the sequence, so the result

$$\nabla c_i(x)^{\mathrm{T}} p = 0 \tag{3.13}$$

holds. On the other hand, if $p$ is tangent to a *nonbinding* sequence, then

$$\nabla c_i(x)^{\mathrm{T}} p > 0. \tag{3.14}$$

As was the case with equality constraints, optimality conditions are determined by examining the behavior of the objective function along feasible paths and sequences.

**Theorem 3.3.1** ([16] Chapter 6)**.** *Assume that $f$ and $c$ are differentiable. A feasible point $x^*$ is a local solution of problem* (3.12) *only if $\nabla f(x^*)^{\mathrm{T}} p \geq 0$ for every $p \in \mathcal{T}(x^*)$, i.e., if $\nabla f(x^*) \in \mathcal{T}^\star(x^*)$.*

It follows from (3.13) and (3.14) that every vector $p \in \mathcal{T}(x^*)$ satisfies $J_a(x^*)p \geq 0$. This relation describes the linearized tangent cone $T_{\scriptscriptstyle L}(x^*)$ of the linearized active constraints $c_a(x; x^*) = c_a(x^*) + J_a(x^*)(x - x^*)$. So, the goal is to discover algebraic conditions that imply that $\nabla f(x^*)^{\mathrm{T}} p \geq 0$ for all $p$ such that $J_a(x^*) \geq 0$. Conditions for the existence of Lagrange multipliers follow directly from *Farkas' lemma*.

**Theorem 3.3.2** (Lagrange Multipliers for (3.12), [16] Chapter 6)**.** *Assume that $f$ and $c$ are differentiable at a feasible point $x^*$. If $J_a(x^*) \in \mathbb{R}^{m_a \times n}$ denotes the matrix of active constraint gradients at $x^*$, then*

1. *$\nabla f(x^*)^{\mathrm{T}} p \geq$ for all $p$ such that $J_a(x^*)p \geq 0$ if and only if $\nabla f(x^*) - J_a(x^*)^{\mathrm{T}} y_a^*$ for some $y_a^* \in \mathbb{R}^{m_a}$ such that $y_a^* \geq 0$.*

2. *There exists a vector $p$ such that$\nabla f(x^*)^{\mathrm{T}} p < 0$ and $J_a(x^*)p \geq 0$ if and only if $\nabla f(x^*) - J_a(x^*)^{\mathrm{T}} y_a \neq 0$ for any $y_a \geq 0$.*

*3. Exactly one of the following statements holds:*

*(a) $\nabla f(x^*)$ can be written as a nonnegative linear combination of the columns of $J_a(x^*)^{\mathrm{T}}$, or*

*(b) there exists a vector $p$ such that $\nabla f(x^*)^{\mathrm{T}} p < 0$ and $J_a(x^*)p \geq 0$.*

The key difference between this and the equality-constrained case is that the Lagrange multipliers are nonnegative, whereas previously they were unconstrained.

**Definition 3.3.6** (First-order KKT point for (3.12), active set form, [16] Chapter 6)**.** The *first-order KKT conditions* for the inequality constrained problem (3.12) hold at $x^*$ if there exists a vector of Lagrange multipliers $y_a^* \in \mathbb{R}^{m_a}$ such that

$$\nabla f(x^*) - J_a(x^*)^{\mathrm{T}} y_a^* = 0, \tag{3.15a}$$

$$c(x^*) \geq 0, \quad \text{and} \quad c_a(x^*) = 0, \tag{3.15b}$$

$$y_a^* \geq 0. \tag{3.15c}$$

This definition is often refined into a second definition in which the vector of Lagrange multipliers is extended to include zero multipliers for the inactive constraints.

**Definition 3.3.7** (First-order KKT point for (3.12), complementarity form, [16] Chapter 6)**.** The *first-order KKT conditions* for the inequality constrained problem (3.12) hold at $x^*$ if there exists a vector of Lagrange multipliers $y^* \in \mathbb{R}^m$ such that

$$\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0, \tag{3.16a}$$

$$c(x^*) \geq 0, \tag{3.16b}$$

$$y^* \geq 0, \tag{3.16c}$$

$$c(x^*) \cdot y^* = 0. \tag{3.16d}$$

Condition (3.16a) can be interpreted as a stationarity condition of the Lagrangian function $L(x, y) = f(x) - y^{\mathrm{T}} c(x)$, i.e., $\nabla_x L(x^*, y^*) = 0$. Depending on the problem, there may be more than one possible value of $y^*$ for a first-order KKT point $x^*$. This motivates the following definition.

**Definition 3.3.8** ([16] Chapter 6)**.** Given a KKT point $x^*$ for problem (3.12), the set of *acceptable Lagrange multipliers* is defined as

$$\mathcal{Y}(x^*) = \{y \in \mathbb{R}^m \ : \ \nabla f(x^*) - J(x^*)^{\mathrm{T}} y = 0, \text{ and } c(x^*) \cdot y^* = 0\}.$$

Condition (3.16d) forces the Lagrange multiplier $y_i$ of an inactive constraint $c_i(x) \geq 0$ to be zero. On the other hand, if the $i$-th constraint is active, there is nothing preventing $y_i^*$ from being zero. *Strict complementarity* is defined to be the case when all active constraints have strictly positive Lagrange multipliers. If an active constraint does not satisfy strict complementarity, i.e., $y_i^\star = 0$ is the only acceptable multiplier, then $c_i(x) \geq 0$ is said to have a *null multiplier*, and is said to be *weakly active*. If there exists a single positive acceptable multiplier, it is said to be *strongly active*.

As was the case with equality-constrained problems, the first-order KKT conditions are not always necessary conditions for optimality. Some constraint qualifications need to hold in order for the KKT conditions to be used as a certificate of optimality. Here, a few of the most commonly used constraint qualifications are enumerated.

**Definition 3.3.9** (Guignard constraint qualification, [16] Chapter 6)**.** The Guignard constraint qualification holds at $x^*$ if $T_L^\star(x^*) = T^\star(x^*)$.

The following constraint qualifications all imply the Guignard constraint qualification.

**Definition 3.3.10** (First-order constraint qualifications, [16] Chapter 6)**.**

1. The *Abadie constraint qualification* holds at $x$ if $x$ is strictly feasible, i.e., no constraints are active, or $\mathcal{T}_L(x) = \mathcal{T}(x)$.

2. The *linear constraint qualification* holds at $x$ if the active constraints at $x$ are all linear.

3. The *linear independence constraint qualification (LICQ)* holds at $x$ if the active constraint gradients at $x$ are all linearly independent.

4. The *Mangasarian-Fromovitz constraint qualification (MFCQ)* holds at $x$ if there exists a vector $p$ such that $\nabla c_i(x)^{\mathrm{T}} p > 0$ for all $i \in \mathcal{A}(x)$.

5. *Slater's condition* holds if the feasible set $\Omega = \{x \ : \ c(x) \geq 0\}$ is convex and there exists a strictly feasible point $x$, i.e., $c(x) > 0$.

The first-order necessary conditions for optimality can now be stated.

**Theorem 3.3.3** ([16] Chapter 6)**.** *If $x^*$ is a local minimizer of problem* (3.2) *at which the MFCQ holds, then $x^*$ is a KKT point.*

*Proof.* Let $x^*$ be a local minimizer at which MFCQ holds. Then $x^* \in \Omega$. If the set of active constraint $\mathcal{A}(x)$ is empty, then $x^*$ is an unconstrained local minimizer, and the result holds. Suppose then at least

one constraint is active. The MFCQ states that there exists a vector $p$ such that $J_a(x^*)p > 0$. As $c$ is differentiable, the Taylor-series of $c$ yields

$$c_i(x^* + \alpha p) = c_i(x^*) + \alpha \nabla c_i(x^*)^{\mathrm{T}} p + \mathcal{O}(||p||^2).$$

. If $\alpha$ is sufficiently small and the $i$-th constraint is active, it follows that $c_i(x^* + \alpha p) > 0$. If, for this same vector $p$, $\nabla f(x^*)^{\mathrm{T}} p < 0$, then the differentiability of $f$ similarly gives $f(x^* + \alpha p) < f(x^*)$, as $p$ would be a descent direction for $f$. This contradicts the fact that $x^*$ is a local minimizer. Therefore, it can be concluded that $\nabla f(x^*)^{\mathrm{T}} p \geq 0$ for every $P$ such that $J(x^*)p > 0$. By Farkas' lemma, $\nabla f(x^*)$ is a nonnegative linear combination of the columns of $J_a(x^*)$. The result follows. □

## Second-Order Optimality Conditions

Like the unconstrained and equality-constrained cases, first-order conditions alone are insufficient to ensure optimality, unless the problem in question is convex. As in the equality-constrained case, second-order conditions involve the Hessian of the Lagrangian function

$$H(x, y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 c_i(x).$$

A key difference between the equality-constrained case and the inequality-constrained case is that in the inequality-constrained case, second-order conditions need not be examined along all feasible directions. Instead, it suffices to investigate the curvature of the objective function $f$ along feasible directions on which $f$ is stationary. In the equality-constrained case, these two sets of directions were identical, whereas here, they are not. It suffices to only examine the curvature of $f$ along *binding* feasible directions, i.e., direction $p$ such that $J_a(x)p = 0$. This motivates a second-order constraint qualification.

**Definition 3.3.11** (Second-order constraint qualification (SOCQ). [16] Chapter 6)**.** A second-order constraint qualification for inequality-constrained optimization problems holds at a feasible point $x$ if every nonzero $p$ satisfying $J_a(x)p = 0$ is tangent to a twice-differentiable path $x(\alpha)$ such that $c_a(x(\alpha)) = 0$ for all $0 \leq \alpha < \widehat{\alpha}$ for some $\widehat{\alpha} > 0$.

The second-order necessary conditions for inequality-constrained optimization can now be stated.

**Theorem 3.3.4** (Second-order necessary conditions for (3.12), [16] Chapter 6)**.** *Let $x^*$ be a point such that $c(x^*) \geq 0$, with $c(x^*)_a = 0$. If the first and second-order constraint qualifications hold at $x^*$, then $x^*$ is a local solution to* (3.12) *only if*

1. *$x^*$ is a first-order KKT point, and*

2. *for some $y \in \mathcal{Y}(x^*)$ and all $p \neq 0$ satisfying $\nabla f(x^*)p = 0$ and $J_a(x^*)p \geq 0$, it holds that $p^{\mathrm{T}} H(x^*, y^*)p \geq 0$.*

There are many different statements of the second-order sufficient conditions for both strict and isolated minimizers of (3.12); too many to discuss here. This discussion is limited to the statement of sufficient second-order conditions for a strict minimizer.

**Theorem 3.3.5** (Second-order sufficient conditions for a strict minimizer (3.12), [16] Chapter 6)**.** *A point $x^*$ is a strict minimizer of problem* (3.12) *if*

1. *$x^*$ is a KKT point, and*

2. *There exists a vector $y \in \mathcal{Y}(x^*)$ such that for all $p \neq 0$ such that $\nabla f(x^*)^{\mathrm{T}} p = 0$ and $J_a(x^*)p \geq 0$, there exists a positive scalar $\omega$ such that $p^{\mathrm{T}} H(x^*, y^*)p \geq \omega ||p||^2$.*

*Proof.* For the same of establishing a contradiction, suppose that the assumptions of the theorem hold, but that $x^*$ is not a strict local minimizer. Then there exists a feasible sequence $\{x_k\} \subseteq \Omega$ that converges to $x^*$ such that $f(x_k) \leq f(x^*)$. Let $\{t_k\}$ and $\{p_k\}$ denote the sequences given by the equation $x_k = x^* + t_k p_k$, where $t_k$ is a normalizing constant such that $||p_k|| = 1$ for all $k$. The sequence $\{p_k\}$ lies in a compact set, and thus $\{p_k\}$ has a convergent subsequence. Therefore, it can be assumed, without loss of generality, that $p_k \to p$ for some vector $p$. Furthermore, due to the fact that a nonnegative $y$ is assumed to exist, it must be true that $c(x_k)^{\mathrm{T}} y \geq 0$.

For each $k$, let $\phi_k(t) = c(x^* t + p_k)^{\mathrm{T}} y$, so that $\phi_k(0) = 0$ and $\phi_k(t_k) = c(x_k)^{\mathrm{T}} y \geq 0$. The Taylor-expansion of $\phi_k$ then gives

$$\phi_k(t_k) = t_k \phi'(0) + \frac{1}{2} t_k^2 \phi_k''(\theta_k t_k) \geq 0 \tag{3.17}$$

for some $\theta_k \in (0, 1)$. The first and second-order derivatives of $\phi$ are

$$\phi_k'(t) = p_k^{\mathrm{T}} J(x^* + t p_k)^{\mathrm{T}} y, \quad \text{and} \quad \phi_k''(t) = \sum_{i=1}^{m} y_i p_k^{\mathrm{T}} \nabla^2 c_i(x^* + t p_k) p_k.$$

Therefore,

$$\phi_k'(0) = p_k^{\mathrm{T}} J(x^*) y. \tag{3.18}$$

By assumption, $f(x^* + t_k p_k) - f(x^*) \leq 0$, thus

$$f(x^* + t_k p_k) - f(x^*) = t_k \nabla f(x^*)^{\mathrm{T}} p_k + \frac{1}{2} p_k^{\mathrm{T}} \nabla f(x^* + \xi_k t_k p_k) p \leq 0, \tag{3.19}$$

for some $\xi_k \in (0, 1)$. Combining (3.17), (3.18), and (3.19) yields

$$t_k \left( \nabla f(x^*) - J(x^*)^{\mathrm{T}} y \right)^{\mathrm{T}} p_k + \frac{1}{2} t_k^2 p_k^{\mathrm{T}} \left( \nabla^2 f(x^* + \xi_k t_k p_k) - \sum_{i=1}^{m} y_i \nabla^2 c_i (x^* + \theta_k t_k p_k) \right) p_k \leq 0.$$

As $x^*$ is assumed to be a KKT point, and $y \in \mathcal{Y}(x^*)$, $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y = 0$. Therefore,

$$p_k^{\mathrm{T}} \left( \nabla^2 f(x^* + \xi_k t_k p_k) - \sum_{i=1}^{m} y_i \nabla^2 c_i (x^* + \theta_k t_k p_k) \right) p_k \leq 0.$$

In the limit as $k \to \infty$,

$$p_k^{\mathrm{T}} H(x^*, y^*) p_k \leq 0. \tag{3.20}$$

Now, as $t_k \to 0$ and $\phi_k(t_k) \geq 0$, (3.17) implies that $\liminf k \to \infty \phi_k'(0) \geq 0$. But $p_k \to p$, and therefore (3.18) implies that $\lim_{k \to \infty}$ exists an is given by

$$\lim_{k \to \infty} \phi'(0)_k = p^{\mathrm{T}} J(x^*) y. \tag{3.21}$$

The assumed properties of $x^*$ then yield

$$\lim_{k \to \infty} \phi'(0)_k = p^{\mathrm{T}} J(x^*) y = \nabla f(x^*)^{\mathrm{T}} p.$$

By the construction of the vector $p$, it holds that

$$\lim_{k \to \infty} \phi'(0)_k = p^{\mathrm{T}} J(x^*) y = \nabla f(x^*)^{\mathrm{T}} p = 0. \tag{3.22}$$

As $c_a(x^*) = 0$ and $c_a(x^* + t_k p_k) \geq 0$, it must hold that $J_a(x^*) p \geq 0$. Along with (3.22), this gives

$$J_a(x^*) p \geq 0 \quad \text{and} \quad \nabla f(x^*)^{\mathrm{T}} p = 0.$$

Combining the above result with (3.20) contradicts the assumptions of the theorem. Therefore, $x^*$ is a strict local minimizer. □

### 3.3.2 Interior-Point Methods

Interior point methods are some of the most widely-used methods for solving nonlinear inequality-constrained optimization problems. In this section, a brief overview of modern approaches to interior methods is presented with the goal of building up enough intuition to motivate the all-shifted primal-dual penalty-barrier method.

The motivating idea behind some of the more basic interior-point methods is to solve problem (3.12) by approximately solving a sequence of unconstrained problems whose solutions converge to the solution of (3.12). This idea is quite similar to the motivating ideas of penalty methods and augmented Lagrangian methods for equality-constrained optimization. The situation is complicated by the fact the constraints in question are inequality constraints. Thus, the objective function of the proposed unconstrained problem should not penalize points that are feasible but should penalize points that are infeasible. This motivates the idea of a *barrier* function, i.e., a function that is defined to be infinite off of the feasible region. Consider the function

$$I_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega \\ \infty & \text{if } x \notin \Omega \end{cases}.$$

If $\Omega$ is convex, this function is sometimes referred to as the *convex indicator function*. Problem (3.12) is then equivalent to solving

$$\min_{x \in \mathbb{R}^n} f(x) + I_\Omega(x). \tag{3.23}$$

Unfortunately, none of the previously discussed methods for solving unconstrained problems can be applied to (3.23), due to the extreme irregularity of the indicator function. The goal then is to approximate (3.23) with a sequence of problems with computable solutions. Consider the function

$$I_{\log}(x) = -\sum_{i=1}^{m} \log(c_i(x)).$$

This is the so-called *logarithmic-barrier function* and is the most commonly used barrier function in interior-point methods. With this function, the composite objective function

$$B(x; \mu^B) = f(x) - \mu^B I_{\log}(x) = f(x) - \mu^B \sum_{i=1}^{m} \log(c_i(x))$$

can be constructed for some positive *barrier parameter* $\mu^B$. Note that $B(x; \mu^B)$ preserves the differentiability

of $f$ and $c$ on the strict interior of the feasible set and is infinite outside the feasible set and on $\{x : c(x) = 0\}$. As $\mu^B$ is driven towards zero, $B(x; \mu^B)$ behaves like $f$ except in close proximity to points where $c(x) = 0$. The classic barrier method is then intuitively given by the following algorithm:

**Algorithm 3.1.** Classical Barrier Algorithm

1: Given $x_0$ such that $c(x_0) > 0$, $\mu_0^B > 0$, and $\gamma_C$ such that $0 < \gamma_C < 1$.
2: $k \leftarrow 0$
3: **while** Not Converged **do**
4: $\quad x_{k+1} \leftarrow \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} B(x; \mu_k^B)$
5: $\quad \mu_{k+1}^B \leftarrow \gamma_C \mu_k^B$
6: $\quad k \leftarrow k + 1$.
7: **end while**

In practical applications of Algorithm 3.1, the unconstrained minimization step is only carried out approximately, with only a few iterations of any given unconstrained method being applied at any given iteration.

Consider the first-order optimality conditions of $B(x; \mu^B)$ for a given $\mu^B$

$$\nabla B(x; \mu^B) = \nabla f(x) - \mu^B \sum_{i=1}^{m} \frac{1}{c_i(x)} \nabla c_i(x). \tag{3.24}$$

Let $y_{\mu^B}$ denote the vector $[y_{\mu^B}]_i = \mu^B / c_i(x)$. Then

$$\begin{aligned} \nabla B(x; \mu^B) &= \nabla f(x) - \mu^B \sum_{i=1}^{m} \frac{1}{c_i(x)} \nabla c_i(x) \\ &= \nabla f(x) - J(x)^{\mathrm{T}} y_{\mu^B}. \end{aligned} \tag{3.25}$$

Furthermore, note that if $c_i(x) > 0$, then $[y_{\mu^B}]_i > 0$. The components of the vector $y_{\mu^B}$ are called the *barrier multipliers*. Observe the similarity between the derivative of $B$ and the first-order stationarity condition $\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* = 0$ for some $y^* \in \mathcal{Y}(x^*)$.

Furthermore, observe the similarity between the complementarity condition $c(x^*) \cdot y^* = 0$ and the following rearrangement of the definition of $y_{\mu^B}$:

$$c(x) \cdot y_{\mu^B} = \mu^B. \tag{3.26}$$

Equation (3.26) is referred to as the *perturbed complementarity* condition. As $\mu^B$ is driven towards zero, the perturbed complementarity condition approaches the true complementarity condition.

It is important to note that the set on which $B(x; \mu^B)$ is less than infinity differs from both the

feasible set $\Omega$ and the topological notion of the interior of $\Omega$, denoted as $\text{int}(\Omega)$.

**Definition 3.3.12** (Strictly Feasible Set)**.** The subset of points in $\Omega$ for which all constraint functions are strictly positive is denoted by $\text{int}_c(\Omega)$ and is given by

$$\text{int}_c(\Omega) = \{x \ : \ c_i(x) > 0\}.$$

If $x \in \text{int}_c(\Omega)$, then $x$ is said to be *strictly feasible*.

Consider the constraint set $\Omega = \{x : x^2 \geq 0\}$. This set is equivalent to $\mathbb{R}$, and therefore $\text{int}(\Omega) = \mathbb{R}$. However, $\text{int}_c(\Omega) = \mathbb{R} \setminus \{0\}$. A constraint for which $\text{int}_c(\Omega) \neq \text{int}(\Omega)$ is said to be *topologically inconsistent*. Such constraints tend to cause problems with interior-point methods, and care should be taken to address topologically inconsistent constraints when modeling problems.

Although the convergence of classical barrier methods can be proven under relatively weak assumptions, such results are not presented here. Instead, the focus is given to a more modern class of methods: the class of primal-dual interior methods.

### 3.3.3 Primal-Dual Interior Methods

The classical barrier method presented in Algorithm 3.1 only solves for the primal variables $x$. The approximation of the Lagrange multipliers, also called the *dual variables*, is inferred via the current iterate $x_k$ and not solved directly. Primal-dual methods instead treat the dual variables similarly to the primal variables. Instead of attempting to iterate towards a constrained minimizer of the objective function $f$, primal methods iterate towards a zero of the first-order KKT conditions by perturbing the complementarity condition, as in (3.26). To be more precise, given a barrier parameter $\mu^B$, a primal-dual method attempts to solve

$$\nabla f(x) - J(x)^{\mathrm{T}} y = 0, \tag{3.27a}$$

$$c(x) \cdot y = \mu^B e. \tag{3.27b}$$

Let $C(x)$ denote the diagonal matrix $\text{diag}(c(x))$, $Y = \text{diag}(y)$, and $F^{\mu^B}$ be

$$F^{\mu^B} = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ C(x)Ye - \mu^B e \end{pmatrix}.$$

Newton's method can be applied directly to $F^{\mu^B}$. The corresponding Newton's equations are given by

$$\begin{pmatrix} H(x,y) & -J(x)^{\mathrm{T}} \\ YJ(x) & C(x) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ C(x)Ye - \mu^B e \end{pmatrix}. \tag{3.28}$$

This matrix can be symmetrized to yield a regularized saddle-point system

$$\begin{pmatrix} H(x,y) & J(x)^{\mathrm{T}} \\ J(x) & -Y^{-1}C(x) \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ C(x)e - \mu^B Y^{-1} e \end{pmatrix}. \tag{3.29}$$

Recall from the discussion of the classical barrier method that the barrier multipliers were defined as $[y_{\mu^B}]_i = \mu^B / c_i(x)$. Use $\pi^y$ to denote this auxiliary vector. Then (3.29) can be written as

$$\begin{pmatrix} H(x,y) & J(x)^{\mathrm{T}} \\ J(x) & -Y^{-1}C(x) \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ (Y^{-1}C(x))(y - \pi^y) \end{pmatrix},$$

or, letting $D = Y^{-1}C(x)$, as

$$\begin{pmatrix} H(x,y) & J(x)^{\mathrm{T}} \\ J(x) & -D \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ D(y - \pi^y) \end{pmatrix}. \tag{3.30}$$

All primal-dual methods begin with this formulation. On its own, however, these equations are insufficient to ensure convergence. For starters, Newton's method is a zero-finding strategy; thus, iterations of this form may not converge to a minimizer. However, if a suitable technique is employed to guarantee a sufficient decrease at each iteration, methods based on solving equation (3.30) perform exceptionally well in practice.

Typically, primal-dual interior methods are broken into inner and outer iterations. Inner iterations correspond to iterations of Newton's method or one of its variants for a particular value of $\mu^B$. Outer iterations correspond to a reduction of the barrier parameter once the inner iterations have made sufficient progress. It remains to show what it means for a sequence of inner iterations to achieve "sufficient progress."

One of the most popular methods to measure progress is to define a suitable merit function. While some methods use the primal log-barrier function $f(x) - \mu^B \sum_{i=1}^m \log(c_i(x))$ as a merit function, this goes against the spirit of the primal-dual method, in which both primal and dual variables are considered at

the same time. Instead, consider the Forsgren-Gill merit function presented in [11]:

$$M(x, y; \mu^B) = f(x) - \mu^B \sum_{i=1}^{m} \left( \log c_i(x) + \log \frac{c_i(x) y_i}{\mu^B} + 1 - \frac{c_i(x) y_i}{\mu^B} \right). \tag{3.31}$$

This merit function penalizes departure from the feasible path, as well as the departure from the *barrier trajectory*, defined to be the path of zeroes of the function $F^{\mu^B}$ for decreasing values of $\mu^B$. An equivalent form of (3.31) (up to constant terms) is

$$M(x, y; \mu^B) = f(x) + c(x)^{\mathrm{T}} y - 2\mu^B \sum_{i=1}^{m} \log c_i(x) - \mu^B \sum_{i=1}^{m} \log y_i. \tag{3.32}$$

In this form, it becomes clear that $M$ penalizes the departure of the feasible set for the primal constraints and the departure of the dual variables from the dual constraint $y \geq 0$. Simple calculation yields

$$\nabla M(x, y; \mu^B) = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}}(2\pi^y - y) \\ D(y - \pi^y) \end{pmatrix} \tag{3.33}$$

and

$$\nabla^2 M(x, y; \mu^B) = \begin{pmatrix} H(x, 2\pi^y - y) + 2J(x)^{\mathrm{T}} C(x)^{-1} \Pi^y J & J(x)^{\mathrm{T}} \\ J(x) & D\Pi^y Y^{-1} \end{pmatrix}. \tag{3.34}$$

By the second part of (3.33), it can be seen that $y$ and $\pi^y$ are equivalent at stationary points of $M$. This motivates the substitution

$$H^M = \begin{pmatrix} H(x, y) + 2J(x)^{\mathrm{T}} D^{-1} J & J(x)^{\mathrm{T}} \\ J(x) & D \end{pmatrix}. \tag{3.35}$$

Linear and quadratic models of the merit function can be formed with $\nabla M$ and $H^M$, which allows line-search and trust-region methods to be applied to $M$. The modified Newton's equations of $M$ are thus

$$\begin{pmatrix} H(x, y) + 2J(x)^{\mathrm{T}} D^{-1} J & J(x)^{\mathrm{T}} \\ J(x) & D \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}}(2\pi^y - y) \\ D(y - \pi^y) \end{pmatrix}.$$

A simple transformation of the above Newton's equations shows that this is equivalent to Newton's equations for $F^{\mu^B}$ (3.30).

### 3.3.4 The Slack Formulation

Consider a line-search approach applied to the merit function defined in (3.31). Suppose a search direction $(\Delta x, \Delta y)$ is computed such that, at iteration $k$, $c(x_k + \Delta x)_i < 0$. Then the merit function $M$ is undefined at $(x_k + \Delta x, y_k + \Delta y)$. In this case, maintaining the strict feasibility of the iterates becomes complicated by the fact that each constraint $c(x_k + \alpha_k \Delta x)$ needs to be checked before evaluating the merit function $M$. Fortunately, a straightforward remedy exists. Each inequality constraint can be reformulated into a linear inequality constraint and a nonlinear equality constraint by introducing a vector $s$ of *slack variables*. To be more precise, for each $i$, the constraint $c_i(x) \geq 0$ becomes $c_i(x) - s_i = 0$ and $s_i \geq 0$. Thus, problem (3.12) becomes

$$\begin{aligned} &\underset{x \in \mathbb{R}^n, s \in \mathbb{R}^m}{\text{minimize}} \; f(x) \\ &\text{subject to } c(x) - s = 0, \quad \text{and} \quad s \geq 0. \end{aligned} \tag{3.36}$$

Primal-dual interior and penalty methods can then be combined to handle the inequality and equality constraints.

# Chapter 4

# The Trust-Region Subproblem

## 4.1 Overview

This section reviews two commonly used methods for solving the trust-region subproblem. Additionally, three novel approaches for solving the trust-region subproblem are introduced. In the most general case, the trust-region subproblem can be expressed as the following quadratically constrained quadratic optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & q(x) = \frac{1}{2} x^{\mathrm{T}} H x + g^{\mathrm{T}} x \\ \text{subject to} \quad & ||x||_M \leq \delta, \end{aligned} \tag{4.1}$$

where $H$, $M \in \mathbb{R}^{n \times n}$, $H = H^{\mathrm{T}}$, $M = M^{\mathrm{T}}$, $M \succ 0$, and $g \in \mathbb{R}^n$. As $M$ is positive definite, many authors use the Cholesky decomposition of $M$, $M = R^{\mathrm{T}} R$, to transform the problem into the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & \frac{1}{2} x^{\mathrm{T}} \bar{H} x + \bar{g}^{\mathrm{T}} x \\ \text{subject to} \quad & ||x||_2 \leq \delta, \end{aligned}$$

to avoid explicitly working with the matrix $M$. While there is nothing theoretically wrong with this approach, there exist situations in which the Cholesky factorization of $M$ would add significant overhead to the run time of any algorithm, so all results shall work with the generalized form. Before analyzing various methods for solving the trust-region subproblem, it is important to understand the conditions guaranteeing that a point $x$ is a solution.

### 4.1.1 Optimality Conditions

Algorithms for finding the global minimum of the trust-region subproblem are based on the following result.

**Theorem 4.1.1** ([16] Chapter 3)**.** *Let $\delta > 0$. A vector $x$ is a global minimum of (4.1) with radius $\delta$ if and only if $||x||_M \leq \delta$ and there exists a scalar $\sigma \geq 0$ such that*

$$(H + \sigma M)x + g = 0, \quad \textit{and} \quad \sigma\big(\tfrac{1}{2}\delta^2 - \tfrac{1}{2}x^{\mathrm{T}}Mx\big) = 0, \tag{4.2}$$

*with $H + \sigma M$ positive definite. If $H + \sigma M$ is positive definite, then the global minimizer is unique.*

*Proof.* The above theorem can be proven using the second-order optimality conditions for constrained minimization. However, this proof will not use such techniques. Suppose that the pair $(x, \sigma)$ satisfy (4.2), and that $||x||_M \leq \delta$ and $H + \sigma M \succeq 0$. Then $x$ is the global minimum of the unconstrained optimization problem $\min_v \mathcal{Q}(v) = g^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}(H + \sigma M)v$. Therefore, for all $v \in \mathbb{R}^n$, $\mathcal{Q}(v) \geq \mathcal{Q}(x)$. Thus,

$$g^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}Hv \geq g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx + \frac{1}{2}\sigma(x^{\mathrm{T}}Mx - v^{\mathrm{T}}Mv). \tag{4.3}$$

Now, $||x||_M = \delta$, and $\sigma \geq 0$, so $g^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}Hv \geq g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx + \frac{1}{2}\sigma(\delta^2 - v^{\mathrm{T}}Mv)$. Thus, $g^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}Hv \geq g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx$ for all $v$ such that $v^{\mathrm{T}}Mv \leq \delta^2$. Therefore, $x$ is a solution to the trust-region subproblem.

Assume now that $x$ is additionally a global solution. If $||x||_M < \delta$, then $x$ is an unconstrained minimizer of the objective function. Paired with $\sigma = 0$, $(x, \sigma)$ satisfies the conditions of Theorem 4.1.1.

Now, if $||x||_M = \delta$, then $x$ must solve the constrained problem $\min\{g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx : \frac{1}{2}x^{\mathrm{T}}Mx = \frac{1}{2}\delta^2\}$. The method of Lagrange multipliers guarantees the existence of a multiplier $\lambda \in \mathbb{R}^n$ such that the gradient of the Lagrangian $L(x, \lambda) = g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx - \frac{1}{2}\lambda(\delta^2 - x^{\mathrm{T}}Mx)$ is zero. Differentiating $L$ gives

$$\nabla L = \begin{pmatrix} g + Hx + \lambda Mx \\ x^{\mathrm{T}}Mx - \delta^2 \end{pmatrix} = 0.$$

Consequently, $\sigma = \lambda$. As $x$ solves the trust-region problem, it follows that $g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx \leq g^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}Hv$ for all $v$ such that $v^{\mathrm{T}}Mv \leq \delta^2$. This implies that

$$g^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}Hv \geq g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx + \frac{1}{2}\sigma(x^{\mathrm{T}}Mx - v^{\mathrm{T}}Mv)$$

for any $v$ such that $v^{\mathrm{T}}Mv = x^{\mathrm{T}}Mx$. Using the condition $(H + \sigma M)x + g = 0$ gives

$$\frac{1}{2}(v - x)^{\mathrm{T}}(H + \sigma M)(v - x) \geq 0.$$

As the direction of the vector $v$ is arbitrary, it follows that $v - x$ is also arbitrary, and $H + \sigma M$ must be positive semidefinite.

To show that $\sigma \geq 0$, observe that the optimality conditions for an unconstrained quadratic function with a Hessian matrix $H + \sigma M$ imply that (4.3) is valid for each $v \in \mathbb{R}^n$. If $\sigma \leq 0$, then this implies that

$$g^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}Hv \geq g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx \;\text{ for all }\; v^{\mathrm{T}}Mv \geq x^{\mathrm{T}}Hx.$$

As $x$ solves the trust-region subproblem, it must be an unconstrained minimizer. Otherwise, if $||x||_M = \delta$, then there must exist a $v$ with $v^{\mathrm{T}}Mv > x^{\mathrm{T}}Mx$ such that $g^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}Hv < g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx$, which contradicts the above bound. If $x$ is an unconstrained minimizer, then the Lagrange multiplier $\sigma$ is 0. Thus, $\sigma \geq 0$. $\square$

An important observation is that only one possible value of $\sigma$ exists, such that $H + \sigma M$ is positive semidefinite and singular. Let $\lambda_n \leq \ldots \leq \lambda_1$ be the generalized eigenvalues of the matrix pencil $(H, M)$. For any $\sigma > -\lambda_n$, $H + \sigma M$ is positive definite and nonsingular, and the solution is given by

$$x(\sigma) = -(H + \sigma M)^{-1}g.$$

The function $\psi(x) = \sqrt{x(\sigma)^{\mathrm{T}}Mx(\sigma)}$ is a well-defined function for all $\sigma > -\lambda_n$.

**Corollary 4.1.1.1** ([16] Chapter 3)**.** *The scalar $\sigma$ exists and is unique.*

*Proof.* The feasible set $\{x : ||x||_M \leq \delta\}$ is compact, implying that a global solution $x$ exists. Theorem 4.1.1 implies that there exists a scalar $\sigma \geq 0$ and a vector $x$ such that

$$(H + \sigma M)x + g = 0 \;\text{ and }\; \sigma(\tfrac{1}{2}x^{\mathrm{T}}Mx - \tfrac{1}{2}\delta^2) = 0,$$

with $H + \sigma M$ positive semidefinite. Let $\bar{\sigma}$ also satisfy the optimality conditions. Without loss of generality, assume that $\bar{\sigma} \geq \sigma \geq -\lambda_n$. If $\bar{\sigma} = -\lambda_n$, then $\bar{\sigma} = \sigma$. Assume that $\bar{\sigma} > -\lambda_n$. Then $H + \bar{\sigma}M$ is positive definite, and $x$ is the unique solution. Therefore, $(H + \sigma M)x + g = 0$ and $(H + \bar{\sigma}M)x + g = 0$. Subtracting the two equations yields

$$(\sigma - \bar{\sigma})Mx = 0.$$

Recall that $M$ is positive definite, so $M$ is nonsingular, thus

$$(\sigma - \bar{\sigma})x = 0.$$

If $x \neq 0$, then $\sigma = \bar{\sigma}$. If $x = 0$, then $x$ is the unconstrained minimizer (recall the assumption that $\delta > 0$), thus $\sigma = \bar{\sigma} = 0$. Therefore, $\sigma$ is unique. □

It has been shown that if $\sigma > -\lambda_n$, the solution to the trust-region subproblem is easily attainable once the value of $\sigma$ is known. If, however, $\sigma = -\lambda_n$, then the problem becomes, computationally, much more challenging to solve. The most difficult subset of such problems is frequently referred to as the hard case.

### 4.1.2 The Hard Case

Consider a solution pair $(x, \sigma)$ of (4.1), where $\sigma = -\lambda_n$. By Theorem 4.1.1, $(H - \lambda_n M)x = -g$, with $H - \lambda_n M$ positive semidefinite and singular. Thus, the vector $g$ must lie in the range of the matrix $H - \lambda_n M$, or equivalently, the orthogonal complement to the null space of $H - \lambda_n M$, i.e., $g$ is orthogonal to the leftmost eigenvector of the matrix pencil $H - \lambda_n M$. Therefore, the solution vector $x$ can be decomposed into two components:

$$x = s + u, \text{ with } s = -(H - \lambda_n M)^{\dagger} g, \text{ and } u \in \text{null}(H - \lambda_n M).$$

Consider now the constraint $||x||_M \leq \delta$. If $\lambda_n < 0$, then the constraint must be active, so $||x||_M = \delta$. Without loss of generality, assume that $\lambda_n$ is a simple eigenvalue. Let $u_n$ be the normalized eigenvector of $\lambda_n$ such that $u_n^{\mathrm{T}} M u_n = 1$. Then $x = s + \alpha u_n$ for some scalar $\alpha$ such that $||x||_M = \delta$. If $s^{\mathrm{T}} M s > \delta^2$, no such $\alpha$ exists. In this case, the vector $s$ already violates the trust-region constraint, so $\sigma \neq -\lambda_n$. Therefore, the hard case occurs when the following two conditions are met:

1. $g \in \text{null}(H - \lambda_n M)$, and
2. $s = -(H - \lambda_n M)^{\dagger} g$ is such that $s^{\mathrm{T}} M s \leq \delta^2$.

The solution of the subproblem for the hard case is then given by

$$x = -(H - \lambda_n) M^{\dagger} g + \alpha u_n, \quad \sigma = -\lambda_n,$$

where $\alpha$ is chosen such that $||x||_M = \delta$.

This degenerate case is called the hard case as it adds the additional complexity of solving an eigenvector problem to solve the trust-region subproblem. Several algorithms for solving the trust-region subproblem disregard the hard case entirely and will converge to some vector that is not necessarily a minimizer. Fortunately, the hard case rarely occurs in practice. On the other hand, the so-called "near hard case" can occur quite frequently. As a result, algorithms that disregard the hard case may struggle to converge if the problem is arbitrarily close to the hard case. This problem is particularly evident with the GLTR algorithm.

## 4.2 The Moré-Sorensen Algorithm

The Moré-Sorensen approach is perhaps the most well-known method for solving the trust-region subproblem. It is relatively straightforward to implement and can quickly find highly accurate solutions in low to medium-dimension problems. In the methods to follow, this algorithm will primarily be used for finding the solutions to the trust-region problem projected onto various subspaces that approximate the true solution, thus representing an essential step in building robust, large-scale methods.

Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of the symmetric definite matrix pencil $(H, M)$, where $H$, $M \in \mathbb{R}^{n\times n}$ are symmetric, and $M$ is positive definite. Let $g \in \mathbb{R}^n$, and $\delta > 0$. Consider the trust-region problem defined by $H$, $M$, $g$, and $\delta$, as in (4.1). Let $x$ and $\sigma$ denote the optimal solution and Lagrange multiplier. If $\sigma > -\lambda_n$, then, by Theorem 4.1.1, $H + \sigma M$ is positive definite, and $x = -(H + \sigma M)^{-1}g$. Use the notation $x(\sigma)$ to denote, for any $\sigma > -\lambda_n$, $-(H + \sigma M)g$. Let $U$ be a matrix whose columns are the $M$-orthonormal generalized eigenvectors of $(H, M)$. Let $V = BU$. Then

$$H = V\Lambda V^{\mathrm{T}} = \sum_{j=1}^{n} \lambda_i v_i v_i^{\mathrm{T}},$$

and

$$(H + \sigma M)^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i + \sigma} u_i u_i^{\mathrm{T}},$$

so

$$x(\sigma) = -\sum_{i=1}^{n} \frac{u_i^{\mathrm{T}} g}{\lambda_i + \sigma} u_i.$$

Therefore, by taking the $M$-norm of each side and squaring,

$$||x(\sigma)||_M^2 = \sum_{i=1}^{n} \frac{(u_i^{\mathrm{T}} g)^2}{(\lambda_i + \sigma)^2}. \tag{4.4}$$

This expression implies that if, for some $i \in \{1, \ldots, n\}$, $u_i^{\mathrm{T}} g \neq 0$, then $||x(\sigma)||_M$ has a pole at $\sigma = -\lambda_i$. Suppose that the solution $x$ lies on the boundary of the trust region, i.e., $||x||_M = \delta$. Then solving the trust-region problem is equivalent to finding the appropriate zero of the function

$$\psi(\sigma) = ||x(\sigma)||_M - \delta, \qquad \text{where} \qquad x = -(H + \sigma M)^{-1} g. \tag{4.5}$$

More specifically, suppose that $\widehat{\sigma}$ is the zero of $\psi$ such that $H + \widehat{\sigma} M$ is positive definite. Then $\sigma = \max\{0, \widehat{\sigma}\}$ is the unique optimal Lagrange multiplier of problem (4.1). A straightforward calculation shows that $\psi$ is convex and nonincreasing on $(-\lambda_n, \infty)$. If $g \neq 0$, then $\psi$ is strictly convex and strictly decreasing.

The following result gives the conditions under which it is guaranteed that a zero of $\psi$ exists in $(-\lambda_n, \infty)$.

**Lemma 4.2.1** ([16] Chapter 3)**.** *If* $\lim_{\sigma \to -\lambda_n} ||x(\sigma)||_M \geq \delta$*, then* $\psi$ *has a unique zero in* $(-\lambda_n, \infty)$*.*

*Proof.* (4.4) implies that $\lim_{\sigma \to \infty} ||x(\sigma)||_M = 0$, and therefore $\lim_{\sigma \to \infty} \psi(\sigma) = -\delta < 0$. If $\lim_{\sigma \to -\lambda_n} ||x(\sigma)||_M > \delta$, then $\lim_{\sigma \to -\lambda_n} \psi(\sigma) > 0$. Therefore, $\psi$ undergoes a sign change on the interval $(-\lambda_n, \infty)$. The result follows as $\psi$ is convex, strictly decreasing, and continuous on $(-\lambda_n, \infty)$. □

Consider the hard-case, where $g \in \text{null}(H - \lambda_n M)^{\perp}$. Then $u_n^{\mathrm{T}} g = 0$. Consider the implications of this in the expansion (4.4) of $||x(\sigma)||_M^2$. If $u_n^{\mathrm{T}} g \neq 0$, then $||x(\sigma)||_M$ diverges as $\sigma \to -\lambda_n$. The case where $g \in \text{null}(H - \lambda_n M)^{\perp}$ is said to be *degenerate*. Note that all instances of the hard case are degenerate, but not all degenerate problems are instances of the hard case.

**Lemma 4.2.2** ([16] Chapter 3)**.** *The quantity* $||x(\sigma)||_M$ *is finite as* $\sigma \to -\lambda_n$ *if an only if* $g \in \text{null}(H - \lambda_n M)^{\perp}$*.*

*Proof.* If $||x(\sigma)||_M$ is finite as $\sigma \to -\lambda_n$, then (4.4) implies that $u_n^{\mathrm{T}} g = 0$ for all $u_i \in \text{null}(H - \lambda_n M)$. This collection of $u_i$ form a basis for $\text{null}(H - \lambda_n M)$, thus $u^{\mathrm{T}} g = 0$ for all $u \in \text{null}(H - \lambda_n M)$, thus $g \in \text{null}(H - \lambda_n M)^{\perp}$. On the other hand, if $||x(\sigma)||_M \to \infty$ as $\sigma \to -\lambda_n$, then $g^{\mathrm{T}} u_i \neq 0$ for at least one $u_i \in \text{null}(H - \lambda_n M)$. □

The following result gives an equivalent characterization for when the trust-region problem is degenerate.

**Lemma 4.2.3** ([16] Chapter 3)**.** *The quantity* $\lim_{\sigma \to -\lambda_n} \psi(\sigma)$ *is finite if and only if the equations* $(H - \lambda_n M)x = -g$ *are compatible, i.e., there exists at least one solution.*

*Proof.* Assume that $\lim_{\sigma\to-\lambda_n}\psi(\sigma)$ is finite. Then Lemma 4.2.2 implies that $u_i^{\mathrm{T}}g = 0$ for all $u_i \in \mathrm{null}(H-\lambda_n M)$, thus $g \in \mathrm{null}(H-\lambda_n M)^{\perp} = \mathrm{range}(H-\lambda_n M)$. Therefore, the system has at least one solution.

On the other hand, if $\lim_{\sigma\to-\lambda_n}\psi(\sigma) = \infty$, then (4.2.2) implies that there exists a vector $u \in \mathrm{null}(H-\lambda_n M)$ such that $u^{\mathrm{T}}g \neq 0$. The vector $g$ can be uniquely decomposed into $g = g_N + g_R$, where $g_N \in \mathrm{null}(H-\lambda_n M)$ and $g_R \in \mathrm{range}(H-\lambda_n M)$. Then it holds that $g^{\mathrm{T}}u = g_N^{\mathrm{T}}u + g_R^{\mathrm{T}}u = g_n^{\mathrm{T}}u \neq 0$, so $g_N \neq 0$. Therefore, $g \neq g_R$, and $g \notin \mathrm{range}(H-\lambda_n M)$, and the system is not compatible. □

**Corollary 4.2.3.1** ([16] Chapter 3)**.** *If* $\lim_{\sigma\to-\lambda_n}\psi(\sigma)$ *is finite, then* $\lim_{\sigma\to-\lambda_n} x(\sigma) = -(H-\lambda_n)^{\dagger}g$

If the trust-region problem is degenerate, and $||(H-\lambda_n M)^{\dagger}g||_M < \delta$, then $\psi$ has no zero in $(-\lambda_n, \infty)$. Only this situation is referred to as the hard case, as outside this situation, the degeneracy of the system does not come into play.

Now, the function $\psi$ could be used in a method for solving the trust-region problem. A safeguarded Newton's method could be applied, and it would eventually converge to a zero, giving a solution on the boundary or terminated early if $\sigma = 0$ yields a solution. However, in most cases, $\psi$ has a singularity at $-\lambda_n$, which may be arbitrarily close to the optimal value of $\sigma$. Newton's method for zero finding converges in one solution if the function is linear but may converge slowly if the function is highly nonlinear, as is the case with $\psi$ near a singularity. Therefore, Newton's method applied $\psi$ will not be a reliable method in many cases.

In [25], Moré and Sorensen instead suggested using Newton's method on the function

$$\phi(\sigma) = \frac{1}{\delta} - \frac{1}{||x(\sigma)||_M}, \quad \text{where} \quad (H+\sigma M)x = -g. \tag{4.6}$$

To be precise, their paper only considered the case $M = I$. However, the result is easily extended to the general case. The derivative of $\psi$ has discontinuities at the eigenvalues of $(-H, M)$ but has no poles, and is approximately linear in a larger neighborhood of the optimal value of $\sigma$, making $\phi$ a much more suitable function on which to apply Newton's method and achieve convergence in a reasonable number of iterations. This forms the basis of the Moré-Sorensen algorithm for solving the trust-region problem. A straightforward calculation shows that the Newton iterates of $\phi$ are given by

$$\sigma_{j+1} = \sigma_j + \frac{x_j^{\mathrm{T}}Mx_j}{x_j^{\mathrm{T}}M(H+\sigma_j M)^{-1}Mx_j}\left(\frac{||x_j||_M - \delta}{\delta}\right), \quad \text{where} \quad (H+\sigma_j M)x_j = -g.$$

Typically, these iterates are implemented by taking the Cholesky decomposition of $H+\sigma_j M$ at

each iteration. These factorizations constitute the majority of the work done by the algorithm. Like any eigenvalue algorithm, the Moré-Sorensen algorithm is iterative, and no exact a priori quantity can be given for the computation needed to solve the trust-region problem. Thus, this algorithm is iterative and approximate, and requires some termination criteria to indicate that the solution found is sufficiently close to the true solution. A reasonable stopping criterion is terminating the algorithm when $|\psi(\sigma)|$ is smaller than some predefined tolerance. However, as $\psi$ will not be exactly zero, the solution $x$ may violate the trust-region constraint by some amount that depends on the given tolerance. To compensate for this, the algorithm instead attempts to find the approximate zero of $\psi$ defined with a scalar $\Delta$ slightly less than the true radius $\delta$. Suppose the Newton iteration terminates when $\sigma$ satisfies

$$(H + \sigma M)x = -g, \quad \text{and} \quad \widehat{\psi}(\sigma) = \frac{\big|||x||_M - \Delta\big|}{\Delta} \leq \varepsilon, \tag{4.7}$$

for some $0 < \varepsilon < 1$. then

$$\Delta(1 - \varepsilon) \leq ||x||_M \leq \Delta(1 + \varepsilon).$$

A reasonable choice of $\Delta$ is $\delta/(1 + \varepsilon)$, as then $||x||_M \leq \delta$, as is required. Thus, the zero finding method should be applied to the perturbed trust-region problem $\min_{x\in\mathbb{R}^n}\{q(x) : ||x||_M \leq \Delta\}$.

**Theorem 4.2.4** ([16] Chapter 3)**.** *Let $\varepsilon$ be a scalar such that $0 < \varepsilon < 1$, $\Delta = \delta/(1 + \varepsilon)$, and let $s \in \mathbb{R}^n$ satisfy*

$$(H + \sigma M)s + g = 0, \quad \textit{and} \quad \big|||s||_M - \Delta\big|/\Delta \leq \varepsilon,$$

*with $H + \sigma M \succeq 0$. Then $s$ satisfies*

$$q(s) \leq (1 - \varepsilon)^2 q(s^*), \quad \textit{where} \quad q(s^*) = \min_{x\in\mathbb{R}^n}\{q(x) \ : \ ||x||_M \leq \Delta\}. \tag{4.8}$$

*Furthermore, $q(s)$ approximates the unique global minimum $q^\star = \min_{x\in\mathbb{R}^n}\{q(x) \ : \ ||x||_M \leq \delta\}$, i.e.,*

$$q(s) \leq \frac{(1 - \varepsilon)^2}{(1 + \varepsilon)^2} q^\star \quad \textit{and} \quad ||s||_M \leq \delta.$$

*Proof.* The vector $s$ is the exact solution to the problem $q(s) = \min_{x\in\mathbb{R}^n}\{q(s) \ : \ ||x||_M \leq ||s||_M\}$. By assumption, $||s||_M \geq (1 - \varepsilon)\Delta \geq ||(1 - \varepsilon)s^*||_M$, thus,

$$q(s) \leq q((1 - \varepsilon)s^*) = (1 - \varepsilon)g^{\mathrm{T}}s^* + \frac{1}{2}(1 - \varepsilon)^2 s^* H s^*.$$

Now, $g^{\mathrm{T}}s^* = -s^*(H + \sigma^* M)s^* \leq 0$, where $\sigma^*$ is the Lagrange multiplier of $s^*$, and therefore

$$q(s) \leq (1-\varepsilon)^2 q(s^*), \quad \text{and} \quad ||s||_M \leq (1+\varepsilon)\Delta.$$

Let $x^*$ be such that $q^\star = q(x^*) = \min_{x \in \mathbb{R}^n}\{q(x) \ : \ ||x||_M \leq \delta\}$. Recall $\delta = (1+\varepsilon)\Delta$. Then $||x^*|| \leq \delta = (1+\varepsilon)\Delta$, so $||x^*/(1+\varepsilon)||_M \leq \Delta$. Then,

$$q(s^*) \leq q(x^*/(1+\varepsilon)) = g^{\mathrm{T}}x^*/(1+\varepsilon) + \frac{1}{2}x^* H x^*/(1+\varepsilon)^2 \leq q(x^*)/(1+\varepsilon)^2.$$

Combining the two results gives

$$q(s) \leq (1-\varepsilon)^2 q(s^*) \leq ((1-\varepsilon)/(1+\varepsilon))^2 q^\star.$$

□

Recall that in the hard case, the solution is given by $x^* = (H - \lambda_n M)^\dagger g + \tau u_n$. In this case, $\psi$ does not have a zero in $(-\lambda_n, \infty)$, and Newton's iteration applied to $\phi$ will not necessarily yield an optimal solution. Simply solving for the eigenvalue $\lambda_n$ is considerably more difficult than finding $\sigma$ in most cases. Therefore, the method should avoid any such explicit calculation unless the dimension of the problem is sufficiently small. The definition of the hard case does, however, suggest an alternative resolution. Consider the case when $\sigma_j$, the current best estimate of the optimal value of $\sigma$, has $\sigma_j > \max\{0, -\lambda_n\}$. Let $s_j$ satisfy $(H + \sigma_j M)s_j + g = 0$. If $||s_j||_M \leq \Delta$, then the method should search for a vector $z_j$ so that the step satisfies $||s_j + z_j|| = \Delta$ and (4.8). Unless $\sigma_j = -\lambda_n$, $z_j$ will not be a null vector. However, if it is sufficiently close to a null vector, then the solution $s_j + z_j$ may be sufficiently accurate as an approximate solution. Let $x = s_j + z_j$. Then,

$$q(x) = -\frac{1}{2}s_j^{\mathrm{T}}(H + \sigma_j M)s_j - \frac{1}{2}\sigma_j \Delta^2 + \frac{1}{2}z_j^{\mathrm{T}}(H + \sigma_j M)z_j.$$

The vector $z$ that minimizes the above quantity is $z = s^* - s_j$. As $s^*$ is unknown, other values of $z_j$ that seek to minimize $q(x)$ should be chosen. A reasonable option would be to find $z_j$ such that $||z_j|| = 1$ and $z_j^{\mathrm{T}}(H + \sigma_j)z_j$ is as small as possible. Then $z_j$ could be scaled so that $x_j = s_j + z_j$ lies on the boundary of the trust region. As $z_j^{\mathrm{T}}(H + \sigma_j M)z_j \geq 0$ for all $z \in \mathbb{R}^n$, including $z_j = s^* - s_j$, it holds that

$$-\frac{1}{2}s_j^{\mathrm{T}}(H + \sigma_j M)s_j - \frac{1}{2}\sigma_j \Delta^2 \leq q(s^*).$$

Suppose that $z_j$ is found to satisfy

$$z^{\mathrm{T}}(H + \sigma_j M)z \leq \varepsilon(2 - \varepsilon)(s_j^{\mathrm{T}}(H + \sigma_j M)x_j + \sigma_j \Delta^2). \tag{4.9}$$

Then

$$q(x) \leq -(1 - \varepsilon)^2(\frac{1}{2}s_j^{\mathrm{T}}(H + \sigma_j M)s_j + \frac{1}{2}\sigma_j \Delta^2) \leq (1 - \varepsilon)^2 q(s^*).$$

Thus, $x = s_j + z_j$ is a sufficiently accurate solution if $z_j$ satisfies (4.9). In their original paper, Moré and Sorensen recommended using LINPACK for finding $z_j$. A more modern implementation might use the LAPACK routine DLACN2, a thread-safe version of DLACON, which estimates the 1-norm condition number of a matrix using matrix-vector products and produces an approximate null vector as a by-product.

The precise termination criteria for the Moré-Sorensen algorithm can now be stated. Let $\{\sigma_j\}$ be the sequence of Lagrange multipliers created by Newton's method applied to $\phi(\sigma)$. $\sigma = \sigma_j$, with associated vector $x_j$ is considered an approximate zero of $\phi(\sigma)$ if $|\widehat{\psi}| \leq \varepsilon$. Additionally, the algorithm is terminated early if either

1. $\sigma_j = 0$ and $\widehat{\psi}(\sigma_j) < \varepsilon$, or
2. $\sigma_j > 0$, $\widehat{\psi}(\sigma_j) < -\varepsilon$, and there exists a sufficiently accurate null vector $z_j$ of $(H + \sigma_j M)$ such that (4.9) is true.

Now, notice that the function $\phi$ on which Newton's method is being performed is univariate. Thus, Newton's method can be combined with a bisection method to create a safeguarded Newton's method. This both improves the performance of the algorithm and guarantees convergence. This involves constructing a sequence of intervals $I_j = [a_j, b_j]$ with $I_j \subset I_{j-1}$ and $\sigma \in I_j$. If any $b_j$ is negative, $\lambda_n < 0$, and the algorithm can be terminated with $\sigma = 0$. If $\sigma_0 \in (-\lambda_n, \infty)$, then all subsequent iterations lie in $(-\lambda_n, \infty)$, and the method converges. For more details, see [25].

## 4.3 The Truncated Conjugate-Gradient Algorithm

The Moré-Sorensen algorithm works very well for problems of small to medium size and even for some large problems in which matrices of the form $H + \sigma M$ are reasonably sparse. However, the need to compute a new factorization of $H + \sigma_j M$ at each iteration makes it unsuitable for large optimization problems. As mentioned, it is important not to waste too much computational effort on solving the trust-region subproblem exactly. One of the first methods for approximately solving the trust-region subproblem that does not rely on explicit factorizations is the truncated conjugate-gradient method. The

method is introduced here as a natural predecessor to the more sophisticated GLTR algorithm and the new locally-optimal preconditioned conjugate-gradient trust-region algorithm presented later. As shown later, phase one of the GLTR algorithm is equivalent to the truncated conjugate-gradient algorithm. Consider an instance of the trust-region problem in which $\sigma = 0$. Then problem (4.1) is equivalent to the unconstrained convex quadratic problem $\min_{x\in\mathbb{R}^n} g^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Hx$. One of the most well-known algorithms for solving problems of this form is the preconditioned conjugate-gradient algorithm, which is given in Algorithm 1.1. As inferred by its name, the truncated conjugate-gradient algorithm is a truncated version of the preconditioned conjugate-gradient method.

More specifically, consider running Algorithm 1.1 on $\min_{x\in\mathbb{R}^n} \frac{1}{2}g^{\mathrm{T}}x + x^{\mathrm{T}}Hx$ using the trust-region matrix $M$ as a preconditioner, i.e., computing the vector $z_k$ via $Mz_k = r_k$. As will be shown in Section 4.5, the iterates $\{x_k\}$ produced by PCG using $M$ as a preconditioner have $||x_k||_M \leq ||x_{k+1}||_M$ for all $k$. Furthermore, the PCG algorithm can detect the positive definiteness of the matrix $H$ by checking that the step-length parameter $\alpha_k$ is positive at each iteration. If, at iteration $k$, it is determined that either $\alpha_k < 0$ or $||x_k||_M > \delta$, then a final update is make to $x_k$, with $x_{k+1} = x_k + \tilde{\alpha}_k p_k$, where $\tilde{\alpha}_k$ is chosen so that $||x_k + \tilde{\alpha}_k p_k||_M = \delta$, and the algorithm is terminated. In the case where $\sigma = 0$, as will be the case when the trust-region method becomes Newton's method, the truncated conjugate-gradient method can find the solution to arbitrary accuracy. Otherwise, the final vector is guaranteed to lie within the trust region. Furthermore, if the algorithm is started with $x_0 = 0$, then the final value $x_k$ is guaranteed to have $q(x_k) \leq q(x^M)$, where $x^M = -\alpha^M B^{-1}g$ and $\alpha^M$ is scalar forcing $x^M$ to lie in the trust-region.

In the case that $\sigma > 0$, this method may produce a result quite far from the true solution to the trust-region problem. For some optimization problems, this may not present an issue. The truncated conjugate-gradient method has been demonstrated to work quite well for a wide range of problems. However, for some problems, it may be necessary that the trust-region subproblem is solved far more accurately at each step. This motivates the GLTR algorithm, which allows the truncated conjugate-gradient algorithm to continue past the point of detecting indefiniteness or exceeding the trust-region boundary at the expense of requiring more iterations.

## 4.4 The GLTR Algorithm

This section presents the generalized Lanczos trust-region (GLTR) algorithm introduced by Gould et al. in [18]. The GLTR algorithm utilizes a Galerkin approach to solving the trust-region problem, i.e., it constructs a matrix $V$ whose columns form a basis for some $m \leq n$ dimensional subspace of $\mathbb{R}^n$, and

problem (4.1) is solved on this subspace. This is equivalent to solving the subproblem

$$\begin{aligned} &\min_{y\in\mathbb{R}^m} \frac{1}{2} y^{\mathrm{T}} V^{\mathrm{T}} H V y + g^{\mathrm{T}} V y \\ &\text{subject to} \quad ||y||_{V^{\mathrm{T}} M V} \leq \delta. \end{aligned} \tag{4.10}$$

By (4.2), a primal dual solution pair $(y, \sigma)$ satisfies

$$V^{\mathrm{T}} g + (V^{\mathrm{T}} H V + \sigma V^{\mathrm{T}} M V) y = 0.$$

The approximate solution $x$ is recovered via $x = Vy$. This is equivalent to the Galerkin condition

$$g + (H + \sigma M)x \perp \text{range}(V), \quad x \in \text{range}(V).$$

In the GLTR algorithm, the basis matrix $V$ is generated via a generalized Lanczos process on the matrix pencil $(H, M)$. The algorithm is terminated when the residual vector $g + (H + \sigma M)x$ is sufficiently close to the zero vector.

### 4.4.1 The Algorithm

As the matrix $M$ in the constraint equation $||x||_M \leq \delta$ is symmetric positive definite, it induces an inner product $\langle x, y \rangle_M = y^{\mathrm{T}} M x$. The GLTR algorithm uses a Lanczos process on the matrix pencil $(H, M)$ to generate a sequence of $M$ orthonormal vectors $\{u_j\}_{j=1}^k$ such that

$$U_j^{\mathrm{T}} H U_j = T_j,$$

where $U_j = [u_1 \cdots u_j]$ and $T_j$ is symmetric and tridiagonal. The choice of the initial vector $u_1$ is given by $\beta = ||g||_{B^{-1}}$ and $u_1 = B^{-1} g / \beta$. Subsequent vectors are computed via the Lanczos decomposition (1.1). These vectors span the $j$ dimensional Krylov subspace

$$\mathcal{K}_j(M^{-1}H, M^{-1}g) = \text{span}\{M^{-1}g, \ldots, (M^{-1}H)^{j-1} M^{-1} g\},$$

just as in the Lanczos-CG method. After each new Lanczos vector is computed, a trust-region subproblem is solved on the subspace $\mathcal{K}_j(M^{-1}H, M^{-1}g)$, i.e.,

$$\begin{aligned} \min_{x\in\mathbb{R}^n} q(x) &= \frac{1}{2}x^{\mathrm{T}}Hx + g^{\mathrm{T}}x \\ \text{subject to} \quad ||x||_M &\leq \delta, \quad x \in \text{range}(U_j). \end{aligned} \tag{4.11}$$

This simplifies to

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \psi(y) &= \frac{1}{2}y^{\mathrm{T}}T_j x + \beta e_1^{\mathrm{T}} y \\ \text{subject to} \quad ||y||_2 &\leq \delta. \end{aligned} \tag{4.12}$$

For any $\sigma$, $T_j + \sigma I$ is tridiagonal. Thus, the Moré-Sorensen algorithm can quickly solve problem (4.12). Unlike Lanczos-CG, a complete Cholesky decomposition of the matrix $T_j + \sigma_j I$ must be computed at each iteration. However, at each iteration, the previous Lagrange multiplier $\sigma_{j-1}$ will be reasonably close to $\sigma_j$, so the Moré-Sorensen algorithm can be easily warm-started using the value of $\sigma_{j-1}$ from the previous iteration. Should $\sigma_j = \sigma_{j+1}$, as will likely be the case close to termination, the Cholesky decomposition of $T_{j-1} + \sigma_{j-1}I$ can be used to find the Cholesky decomposition of $T_j + \sigma_j I$. The projected solution of the original problem is then given by $x = U_j y_j$, where $y_j$ is the solution to problem (4.12). It is crucial to note that this vector $x$ need not be constructed at each iteration but only after convergence is achieved. Further note that at each iteration $j$, the optimality condition $y_j^{\mathrm{T}} U_j^{\mathrm{T}} M U_j y_j = x_j^{\mathrm{T}} M x_j \leq \delta^2$ is satisfied.

Convergence of the GLTR algorithm is measured via the residual vector $r_j = (H + \sigma_j M)U_j y_j + g$. It can be shown that

$$||HU_j y_j + \sigma_j M U_j y_j + g||^2_{M^{-1}} = (\beta_{j+1} e_j^{\mathrm{T}} y_j)^2.$$

Once this measure of the residual is sufficiently small, the vector $x$ is constructed by either regenerating the Lanczos vectors or calling them from in-memory storage.

While the GLTR method performs reasonably well for many large-scale sparse problems, several problems exist. First, the method as described does not account for the hard case. If $g \perp \text{null}(H - \lambda_n M)$, then each Lanczos vector $u_i$ will lie in the $M$-orthogonal compliment of $\text{null}(H - \lambda_n M)$. Thus, the hard case will not be detected, and the iterates $x_j$ will instead converge to the solution of

$$\begin{aligned} \min_{x\in\mathbb{R}^n} &\frac{1}{2}x^{\mathrm{T}}Hx + g^{\mathrm{T}}x \\ \text{s.t } ||x||_M &\leq \delta, \quad x^{\mathrm{T}}Mu = 0 \text{ for all } u \in \text{null}(H - \lambda_n M), \end{aligned} \tag{4.13}$$

completely ignoring the direction with the most negative curvature.

A second issue concerns the rate of convergence. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ be the generalized eigenvalues of the matrix pencil $(H, M)$. This matrix pencil is symmetric definite, and therefore all eigenvalues are real. It is shown in [23] that the rate of convergence for $x_j = U_j y_j$ to the optimal solution $x^*$ is related to the $M$-condition number of the matrix $H + \sigma^* M$ in a manner similar to the preconditioned conjugate-gradient method. Unfortunately, $M$ may not be a suitable preconditioner for the matrix $H + \sigma^* M$. Denote the $M$ condition number as

$$\kappa_M(H + \sigma M) = \frac{\lambda_1 + \sigma}{\lambda_n + \sigma}.$$

Then the following theorem applies.

**Theorem 4.4.1** (GLTR Convergence, [23] Theorem 4.11)**.** *Let $H, M \in \mathbb{R}^{n \times n}$ be symmetric, and $M \succ 0$. Let $g \in \mathbb{R}^n$, and $\delta > 0$. Let $x^*$ and $\sigma^*$ be the optimal solution and Lagrange multiplier, respectively, of the trust-region problem given by $H$, $M$, $g$, and $\delta$, as in* (4.1)*. Let $\{(x_k, \sigma_k)\}$ be the sequence of iterates produced by the GLTR algorithm. Then*

$$||x_k - x^*||_{H+\sigma^* M} \leq 2 \left( \frac{\sqrt{\kappa_M(H + \sigma^* M)} - 1}{\sqrt{\kappa_M(H + \sigma^* M)} + 1} \right)^{k+1} ||x^*||_{H+\sigma^* M}, \tag{4.14}$$

*and*

$$|\sigma^* - \sigma_k| \leq c \left( \frac{\sqrt{\kappa_M(H + \sigma^* M)} - 1}{\sqrt{\kappa_M(H + \sigma^* M)} + 1} \right)^{2(k+1)}$$

*for some constant $c$.*

Suppose $\kappa_M(H + \sigma^* M)$ is sufficiently large. In that case, the GLTR algorithm will require many iterations to converge, so much so that in a practical setting, the algorithm may need to end early before the convergence criteria are reached. It is well known that the Lanczos algorithm is numerically unstable, eventually leading to the vectors $u_j$ losing their $M$-orthogonality. Partial or full reorthogonalization can be implemented. However, these schemes may significantly slow down runtime, particularly as the number of iterations grows very large. Furthermore, they require that the vectors $u_j$ be stored in fast memory, which may be quickly exhausted as the number of iterations grows.

Even if $M$ is a reasonable choice of the preconditioner for $H + \sigma^* M$, the conditioning of the trust-region problem is adversely affected by the proximity of $\sigma^*$ to $-\lambda_n$.

**Lemma 4.4.2.** *Let $H, M \in \mathbb{R}^n$ be symmetric, and $M$ be positive definite. Let $\lambda_1$ and $\lambda_n$ denote the*

*algebraically largest and smallest eigenvalues of* $(H, M)$*, respectively. Let* $\varepsilon > 0$*, and let* $\sigma \geq -\lambda_n$*. If*

$$\sigma + \lambda_n \leq \varepsilon(\lambda_1 - \lambda_n),$$

*then the condition number* $\kappa_M(H + \sigma M)$ *is bounded below by* $1/\varepsilon$

*Proof.* It is straightforward to see that, under the above assumptions,

$$\kappa_M(H + \sigma M) = \frac{\lambda_1 + \sigma}{\lambda_n + \sigma} \geq \frac{\lambda_1 - \lambda_n}{\lambda_n + \sigma} \geq \frac{\lambda_1 - \lambda_n}{\varepsilon(\lambda_1 - \lambda_n)} = \frac{1}{\varepsilon}.$$

□

Unfortunately, the trust-region subproblem must be solved to know whether this lemma applies. However, it can be used to show a priori that specific trust-region problems are poorly conditioned.

**Theorem 4.4.3.** *Let* $H, M \in \mathbb{R}^n$ *be symmetric, and* $M$ *be positive definite. Let* $g \in \mathbb{R}^n$*, and* $\delta > 0$*. Let* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ *denote the eigenvalues of the matrix pencil* $(H, M)$*. Assume that* $\lambda_n < 0$*. Let* $1 > \varepsilon > 0$*. If*

$$\frac{||g||_{M^{-1}}}{\delta} \leq \varepsilon(\lambda_1 - \lambda_n),$$

*then the trust-region problem defined by* $H$*,* $M$*,* $g$*, and* $\delta$ *has an optimal dual solution* $\sigma^*$ *such that the condition number* $\kappa_M(H + \sigma^* M)$ *is bounded below by* $1/\varepsilon$*.*

*Proof.* Consider the optimality condition $(H + \sigma M)x = -g$. Then

$$\begin{aligned} ||g||_{M^{-1}} &= ||(H + \sigma M)x||_{M^{-1}} \\ &\leq ||H + \sigma M||_M ||x||_M \\ &= (\lambda_1 + \sigma)\delta, \end{aligned}$$

so $||g||_{M^{-1}}/\delta - \lambda_1 \leq \sigma$. Similarly, it can be shown that $\sigma \leq ||g||_{M^{-1}}/\delta - \lambda_n$. By the second-order optimality condition $H + \sigma M \succeq 0$, it also holds that $-\lambda_n \leq \sigma$. In addition to the nonnegativity of $\sigma$, the following bounds on $\sigma$ hold:

$$\max\{0, -\lambda_n, \frac{||g||_{M^{-1}}}{\delta} - \lambda_1\} \leq \sigma \leq \max\{0, \frac{||g||_{M^{-1}}}{\delta} - \lambda_n\}. \tag{4.15}$$

By assumption, $||g||_{M^{-1}}/\delta \leq \lambda_1 - \lambda_n$, and $-\lambda_n \geq 0$. Therefore, the maximum on the left-hand side is achieved by $-\lambda_n$. By the upper bound, it also holds that $\sigma + \lambda_n \leq ||g||_{M^{-1}}/\delta$. By Lemma 4.4.2, it

is known that if $\sigma + \lambda_n \leq \varepsilon(\lambda_1 - \lambda_n)$, the lower bound is true. Thus, if $||g||_{M^{-1}}/\delta \leq \varepsilon(\lambda_1 - \lambda_n)$, the condition number is at least $1/\varepsilon$. □

Experiments reveal that this can result in needing a number of iterations on the scale of the dimension of the problem, drastically reducing the reliability of the GLTR method. A similar issue involving the ill-conditioning of trust-region problem occurs in the near hard case.

**Theorem 4.4.4.** *Let $H, M \in \mathbb{R}^n$ be symmetric, and $M$ be positive definite. Let $g \in \mathbb{R}^n$, and $\delta > 0$. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ denote the eigenvalues of the matrix pencil $(H, M)$, and $q_1, \ldots q_n$ denote the corresponding $M$-orthonormal eigenvectors. Assume that $\lambda_n < 0$, and that $|q_n^{\mathrm{T}} g| > 0$. Let $1 > \varepsilon > 0$. Let $\bar{x} = -(H - \lambda_n M)^{\dagger} g$. If*

$$||\bar{x}||_M < \delta \quad \textit{and} \quad |q_n^{\mathrm{T}} g| \leq \varepsilon(\lambda_1 - \lambda_n)(\delta^2 - \bar{x}^{\mathrm{T}} M \bar{x})^{1/2},$$

*then the trust-region problem defined by $H$, $M$, $g$, and $\delta$ has an optimal dual solution $\sigma^*$ such that condition number $\kappa_M(H + \sigma^* M)$ is bounded below by $1/\varepsilon$.*

*Proof.* Let $x^*$ denote the optimal solution $x^* = -(H + \sigma^* M)^{-1} g$. Let $Q$ denote the matrix whose $i$-th column is $q_i$, for $i = 1, \ldots, n$. Then $Q^{\mathrm{T}} M Q = I$, $Q Q^{\mathrm{T}} M = I$, and $M Q Q^{\mathrm{T}} = I$. The vectors $g$, $x^*$, and $\bar{x}$ can each be expanded in terms of this $M$-orthonormal bases:

$$\begin{aligned}
g &= MQQ^{\mathrm{T}} g = \sum_{i=1}^{n} (q_i^{\mathrm{T}} g) M q_i, \\
x^* &= -(H + \sigma^* M)^{-1} g \\
&= -QQ^{\mathrm{T}} M (H + \sigma^* M)^{-1} MQQ^{\mathrm{T}} g \\
&= -Q(\Lambda + \sigma^* I) Q^{\mathrm{T}} g = -\sum_{i=1}^{n} \frac{q_i^{\mathrm{T}} g}{\lambda_i + \sigma^*} q_i, \quad \text{and} \\
\bar{x} &= -(H - \lambda_n M)^{\dagger} g = -QQ^{\mathrm{T}} M (H - \lambda_n M)^{\dagger} MQQ^{\mathrm{T}} g = \sum_{i=1}^{n-1} \frac{q_i^{\mathrm{T}} g}{\lambda_i - \lambda_n} q_i.
\end{aligned}$$

As the matrix $H$ is indefinite, the trust-region constraint $(x^*)^{\mathrm{T}} M x^* \leq \delta^2$ must be active, therefore

$$\delta^2 = \sum_{i=1}^{n-1} \frac{(q_i^{\mathrm{T}} g)^2}{(\sigma^* + \lambda_i)^2} + \frac{(q_n^{\mathrm{T}} g)^2}{(\sigma^* + \lambda_n)^2}.$$

Without loss of generality, assume that $\lambda_{n-1} > \lambda_n$. As $\sigma^* > -\lambda_n$ and $\lambda_i - \lambda_n > 0$ for all $i = 1, \ldots, n-1$,

$$\delta^2 \leq \sum_{i=1}^{n-1} \frac{(q_i^{\mathrm{T}} g)^2}{(\lambda_i - \lambda_n)^2} + \frac{(q_n^{\mathrm{T}} g)^2}{(\sigma^* + \lambda_n)^2},$$

or equivalently

$$(\sigma^* + \lambda_n)^2 \leq \frac{(q_n^{\mathrm{T}} g)^2}{\delta^2 - \sum_{i=1}^{n-1} \frac{(q_i^{\mathrm{T}} g)^2}{(\lambda_i - \lambda_n)^2}} = \frac{(q_n^{\mathrm{T}} g)^2}{\delta^2 - \bar{x}^{\mathrm{T}} M \bar{x}}.$$

Thus, by Lemma 4.4.2, if

$$|q_n^{\mathrm{T}} g| \leq \varepsilon(\lambda_1 - \lambda_n)(\delta^2 - \bar{x}^{\mathrm{T}} M \bar{x})^{1/2},$$

then $\kappa_M(H + \sigma M) \geq 1/\varepsilon$. □

These two results show that unlike the preconditioned-conjugate-gradient algorithm, where a preconditioner can be specifically chosen to reduce the condition number, the trust-region subproblem, when solved with GLTR, has an intrinsic condition number determined by the parameters of the problem itself. If this condition number is large enough, GLTR may take far too many iterations to be considered useful in a practical setting, particularly when many trust-region subproblems must be solved without any prior information.

## 4.5 The Shifted and Inverted GLTR Algorithm

As the previous section shows, using the GLTR method with the matrix pencil $(H, M)$ does not necessarily yield favorable results. Therefore, this section presents a new approach that utilizes a Lanczos process with a shift and invert technique capable of resolving the issues inherent in the GLTR method. For this reason, it is referred to as the shifted and inverted generalized Lanczos trust-region algorithm (SIGLTR).

Let $\mu \geq 0$ be a shift such that $H + \mu M$ is positive definite. If the trust-region problem is not an instance of the hard case, then the ideal choice would be $\mu = \sigma$. The Lanczos process is performed on the symmetric definite matrix pencil $(M, H + \mu M)$, i.e., a sequence of vectors $u_1, \ldots, u_k$ is found such that if $U_k = \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix}$,

$$U_k^{\mathrm{T}} M U_k = T_k, \quad \text{and} \quad U_k^{\mathrm{T}} (H + \mu M) U_k = I, \tag{4.16}$$

where $T_k$ is a $k \times k$ tridiagonal matrix, just as in the GLTR algorithm. Thus, it holds that

$$U_k^{\mathrm{T}} H U_k = U_k^{\mathrm{T}} (H + \mu M) U_k - \mu U_k^{\mathrm{T}} M U_k = I - \mu T_k. \tag{4.17}$$

If the Lanczos process is allowed to run for $n$ iterations, the result is an $H+\mu M$-orthonormal basis matrix $U_n$ for $\mathbb{R}^n$ such that $U_n^{\mathrm{T}} M U_n = T_n$. Therefore, $BU_n = (H+\mu M)U_n T_n$, with

$$T_n = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix}.$$

Equating the columns of both sides gives the following 3-term recursion relation:

$$Mu_j = \beta_j(H+\mu M)u_{j-1} + \alpha_j(H+\mu M)u_j + \beta_{j+1}(H+\mu M)u_{j+1},$$

or, if $(H+\mu M)u_j = v_j$ for all $j$, and $V_j = \begin{bmatrix} v_1 & \cdots & v_j \end{bmatrix}$,

$$Mu_j = \beta_j v_{j-1} + \alpha_j v_j + \beta_{j+1} v_{j+1}.$$

As, by definition, $u_i^{\mathrm{T}} v_j = 0$ for all $i \neq j$, it holds that

$$\alpha_j = u_j^{\mathrm{T}} M u_j = u_j^{\mathrm{T}}(Mu_j - \beta_j v_{j-1}) = u_j^{\mathrm{T}} w_j.$$

Then

$$\beta_{j+1} v_{j+1} = w_j - \alpha_j v_j, \quad ||v_{j+1}||_{(H+\mu M)^{-1}} = 1,$$

so

$$\beta_{j+1} = ||w_j - \alpha_j v_j||_{(H+\mu M)^{-1}},\ v_{j+1} = (w_j - \alpha_j v_j)/\beta_{j+1},\ u_{j+1} = (H+\mu M)^{-1} v_{j+1}.$$

The following steps summarize one iteration of the Lanczos process:

1. $w \leftarrow Mu_j - \beta_j v_{j-1}$
2. $\alpha_j \leftarrow u_j^{\mathrm{T}} w$
3. $w \leftarrow w - \alpha_j v_j$
4. $u_{j+1} \leftarrow (H+\mu M)^{-1} w$
5. $\beta_{j+1} \leftarrow (w^{\mathrm{T}} u_{j+1})^{1/2}$

6. $v_{j+1} \leftarrow w/\beta_{j+1}$

7. $u_{j+1} \leftarrow u_{j+1}/\beta_{j+1}$,

where $\beta_1 = ||g||_{(H+\mu M)^{-1}}$, $v_0 = 0$, $v_1 = -g/\beta_1$, and $u_1 = (H + \mu M)^{-1} v_1$. The Lanczos decomposition can then be written as

$$MU_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^{\mathrm{T}}. \tag{4.18}$$

By applying a similar argument to (4.17), it holds that

$$HU_k = V_k - \mu V_k T_k - \mu \beta_{k+1} v_{k+1} e_k^{\mathrm{T}}. \tag{4.19}$$

At each iteration, a subproblem of the form

$$\begin{aligned} &\min_{y \in \mathbb{R}^k} \frac{1}{2} y^{\mathrm{T}} (I - \mu T_k) y + \beta_1 e_1^{\mathrm{T}} y \\ &\text{subject to} \quad ||y||_{T_k} \leq \delta \end{aligned} \tag{4.20}$$

is solved via the Moré-Sorensen algorithm. The approximate solution to the original problem (4.1) is then given by $x_k = U_k y_k$. As in GLTR, it is unnecessary to construct $x_k$ at each iteration. Instead, $x_k$ is only constructed once the residual vector $r_k = (H + \sigma_k M) x_k + g$ is sufficiently small. By Theorem 4.1.1, the solution $(y_k, \sigma_k)$ to (4.20) has $(I_k + (\sigma_k - \mu_k) T_k) y_k + \beta_1 e_1 = 0$. By applying (4.18) and (4.19), it holds that

$$\begin{aligned} r_k &= (H + \sigma_k M) x_k + g = \\ &= (H + \sigma_k M) U_k y_k + \beta_1 (H + \mu M) U_k e_1 \\ &= (H + \mu M) U_k y_k + (\sigma_k - \mu) M U_k y_k + \beta_1 (H + \mu M) U_k e_1 \\ &= (H + \mu M) U_k y_k + (\sigma_k - \mu)(H + \mu M)(U_k T_k + \beta_{k+1} u_{k+1} e_k^{\mathrm{T}}) y_k + \beta_1 (H + \mu M) U_k e_1 \\ &= (H + \mu M) U_k ((I_k + (\sigma_k - \mu) T_k) y_k + \beta_1 e_1) + (\sigma_k - \mu) \beta_{k+1} (H + \mu M) u_{k+1} e_k^{\mathrm{T}} y_k \\ &= (\sigma_k - \mu) \beta_{k+1} (H + \mu M) u_{k+1} e_k^{\mathrm{T}} y_k. \end{aligned}$$

Thus,

$$r_k^{\mathrm{T}} (H + \mu M)^{-1} r_k = (\sigma_k - \mu)^2 \beta_{k+1}^2 (e_k^{\mathrm{T}} y_k)^2. \tag{4.21}$$

The solution $x_k$ is only constructed once $||r_k||_{(H+\mu M)^{-1}}$ is less than some predefined tolerance. Note that if $\mu = \sigma$, the solution is found after only a single iteration. Intuitively, this method will require far fewer

iterations than the standard GLTR method.

As mentioned, if the optimal Lagrange multiplier $\sigma$ is 0, the GLTR algorithm reduces to Lanczos-CG. The shifted-and-inverted GLTR algorithm can similarly use the matrices $T_k$ to construct conjugate-gradient-like iterates so that $x_k$ is constructed as the algorithm proceeds instead of all at once at the end. At each iteration, the solution to $Hx = -g$ is approximated on the subspace spanned by the first $k$ Lanczos vectors. A Galerkin condition is imposed so that the approximate solution $x_k$ is given by

$$U_k^{\mathrm{T}} H U_k y_k = -U_k^{\mathrm{T}} g = -\beta_1 e_1.$$

By 4.19, this is simply

$$(I_k - \mu T_k) y_k = -\beta_1 e_1. \tag{4.22}$$

At each iteration, $I - \mu T_k$ must be checked for positive definiteness. If indefiniteness is detected, the method switches to solving (4.20) at each iteration.

Positive definiteness is most easily checked by taking a square root-free Cholesky decomposition

$$L_k D_k L_k^{\mathrm{T}} = I_k - \mu T_k,$$

where

$$L_k = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_k & 1 \end{pmatrix} \quad \text{and} \quad D_k = \begin{pmatrix} d_1 & & & & \\ & d_2 & & & \\ & & d_3 & & \\ & & & \ddots & \\ & & & & d_k \end{pmatrix},$$

where $d_1 = 1 - \mu\alpha_1$, $l_k = -\mu\beta_k / d_{k-1}$, and $d_k = (1 - \mu\alpha_k) - l_k d_{k-1} l_k = (1 - \mu\alpha_k) + \mu\, l_k \beta_k$. Equation (4.22) is now separated into two systems

$$L_k z_k = -\beta_1 e_1, \quad D_k L_k^{\mathrm{T}} y_k = z_k.$$

The vector $z_k$ is easily solved as

$$z_k = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_k \end{pmatrix} = \begin{pmatrix} -\beta_1 \\ -l_2\xi_1 \\ -l_3\xi_2 \\ \vdots \\ -l_k\xi_{k-1} \end{pmatrix}.$$

Instead of solving for $y_k$, $x_k$ is directly formed by defining the matrix $P_k$ by $U_k = P_k D_k L_k$. Then

$$x_k = U_k y_k = P_k D_k L_k y_k = P_k z_k.$$

As the first $k-1$ components of $z_k$ are just $z_{k-1}$, $x_k$ can be built via the equation $x_k = P_k z_k = P_{k-1} z_{k-1} + p_k \xi_k = x_{k-1} + p_k \xi_k$. To find $p_k$, take $U_k = P_k D_k L_k$, or

$$\begin{pmatrix} u_1 & u_2 & \cdots & u_k \end{pmatrix} = \begin{pmatrix} p_1 & p_2 & \cdots & p_k \end{pmatrix} \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_k \end{pmatrix} \begin{pmatrix} 1 & l_2 & & \\ & 1 & l_3 & \\ & & \ddots & l_k \\ & & & 1 \end{pmatrix}.$$

Thus,

$$p_1 = (1/d_1)u_1 \quad \text{and} \quad p_k = (1/d_k)(u_k + \mu\beta_k p_{k-1}).$$

This process is repeated at each iteration $k$ until either $d_k \leq 0$ or $||x_k||_M \geq \delta$. The quantity $x_k^{\mathrm{T}} M x_k$ must be calculated recursively at each iteration. Using $x_k = x_{k-1} + p_k \xi_k$ yields

$$\begin{aligned} x_k^{\mathrm{T}} M x_k &= (x_{k-1} + p_k\xi_k)^{\mathrm{T}} M (x_{k-1} + p_k \xi_k) \\ &= x_{k-1}^{\mathrm{T}} M x_{k-1} + 2\xi_k p_k M x_{k-1} + \xi_k^2 p_k^{\mathrm{T}} M p_k, \end{aligned} \tag{4.23}$$

so two additional recursions are required, one for $p_k^{\mathrm{T}} M x_{k-1}$ and another for $p_k^{\mathrm{T}} M p_k$.

$$\begin{aligned} p_k^{\mathrm{T}} M x_{k-1} &= \frac{1}{d_k}(u_k + \mu\beta_k p_{k-1})^{\mathrm{T}} M (x_{k-2} + p_{k-1}\xi_{k-1}) \\ &= \frac{1}{d_k}(u_k^{\mathrm{T}} M x_{k-2} + u_k^{\mathrm{T}} M p_{k-1}\xi_{k-1} + \mu\beta_k (p_{k-1}^{\mathrm{T}} M x_{k-2} + p_{k-1}^{\mathrm{T}} M p_{k-1}\xi_{k-1})). \end{aligned} \tag{4.24}$$

Now, $x_{k-2} = U_{k-2}y_{k-2}$, so $u_k^{\mathrm{T}} M x_{k-2} = 0$. Furthermore, $u_k^{\mathrm{T}} M p_{k-1} = \beta_k / d_{k-1}$. Thus

$$p_k^{\mathrm{T}} M x_{k-1} = \frac{1}{d_k}\left(\frac{\beta_k \xi_{k-1}}{d_{k-1}} + \mu\beta_k(p_{k-1}^{\mathrm{T}} M x_{k-2} + p_{k-1}^{\mathrm{T}} M p_{k-1}\xi_{k-1})\right), \tag{4.25}$$

so all that remains is the recursion relation for $p_k^{\mathrm{T}} M p_k$.

$$\begin{aligned} p_k^{\mathrm{T}} M p_k &= \frac{1}{d_k^2}(u_k + \mu\beta_k p_{k-1})^{\mathrm{T}} M (u_k + \mu\beta_k p_{k-1}) \\ &= \frac{1}{d_k^2}(u_k^{\mathrm{T}} M u_k + 2\mu\beta_k p_{k-1}^{\mathrm{T}} M u_k + \mu^2\beta_k^2 p_{k-1}^{\mathrm{T}} M p_{k-1}) \\ &= \frac{1}{d_k^2}(\alpha_k + 2\mu\frac{\beta_k^2}{d_{k-1}} + \mu^2 p_{k-1}^{\mathrm{T}} M p_{k-1}). \end{aligned} \tag{4.26}$$

These three recursion relations can compute $x_k^{\mathrm{T}} M x_k$ without calculating additional matrix-vector operations. Much like in the Lanczos-CG algorithm, the norm of the residual vector $||g + Hx_k||^2_{(H+\mu M)^{-1}}$ is given by $\xi_{k+1}^2$.

A crucial aspect of the GLTR algorithm is that $x_k^{\mathrm{T}} M x_k$ increases at each iteration while in phase one. This enables the algorithm to switch to phase two upon detecting that the trust-region constraint is violated and detecting indefiniteness. It remains to be seen that the same holds for SIGLTR. Thus, it suffices to investigate the $\xi_k p_k^{\mathrm{T}} M x_{k-1}$ term. By construction, each $\beta_k \geq 0$ and $d_k > 0$. Thus, each $l_k = -\mu\beta_k / d_{k-1} \leq 0$. The scalar $\xi_1$ satisfies $\xi_1 = -\beta_1 \leq 0$, therefore $\xi_i = -l_i\xi_{i-1} \leq 0$ for all subsequent iterations $i$. Thus, each $\xi_k = -l_k\xi_{k-1} \leq 0$. So, it suffices to show that each $p_k M x_{k-1} \leq 0$ to show that $x_k^{\mathrm{T}} M x_k$ is increasing.

$x_0$ is always taken to be the zero vector, so $p_1^{\mathrm{T}} M x_0 = 0$. Thus,

$$p_2^{\mathrm{T}} M x_1 = \frac{1}{d_1}\left(\frac{\beta_2\xi_1}{d_1} + \mu\beta_2 p_1^{\mathrm{T}} M p_1 \xi_1\right) \leq 0.$$

Using induction, it is straightforward to see that this term will always be non-positive. Putting everything together, it is seen that $x_k^{\mathrm{T}} M x_k$ is non-decreasing at each iteration, and therefore, SIGLTR can utilize the same two-phase structure as GLTR. This gives all the necessary elements of the shifted-and-inverted GLTR algorithm.

The key detail is that this still requires at least one factorization of a matrix of the form $(H + \mu M)$. In some cases, this is comparable to the work that GLTR must perform, as GLTR requires a factorization of the positive definite matrix $M$. However, most practical instances of the trust-region problem possess a matrix $M$ that is simple to invert, such as the identity matrix or a diagonal matrix. Thus, the significantly

fewer iterations are the primary benefit of shifting and inverting. Conversely, this method requires far fewer factorizations than a Moré-Sorensen style algorithm, particularly in a nonlinear optimization algorithm, where the choice of $\mu$ can be selected as the dual solution $\sigma$ of the previous problem. Furthermore, the choice of $\mu$ and the factorization of $(H + \mu M)$ can be reused on subsequent solutions of the trust-region subproblem with a restricted radius.

Recall from Section 2.5 the convergence of a trust-region method requires that the computed solution yields a value of the objective function that is less than a fixed fraction of the value of the objective function for some steepest-descent direction. Consider the initial vector Lanczos vector $\beta_1 u_1 = (H + \mu M)^{-1} g$. This vector is colinear with the steepest-descent direction induced by the $H + \mu M$ norm. Thus, the objective value after the first iteration is sufficient to ensure the convergence of an outer trust-region method. Furthermore, the sequence of iterates $\{x_k\}$ generated by SIGLTR satisfies $q(x_{k+1}) \leq q(x_k)$ for all iterations $k$. Thus, solutions computed by SIGLTR are sufficient to ensure the convergence of a trust-region method after one iteration.

### 4.5.1 The Projected Trust-Region Subproblem

Consider the reduced trust-region subproblem (4.20). By Theorem 4.1.1, the optimality conditions of this problem are:

1. $(I + (\sigma - \mu)T_k)y + \beta e_1 = 0$,
2. $\delta^2 - y^{\mathrm{T}} T_k y \geq 0$,
3. $\sigma \geq 0$,
4. $\sigma(\delta^2 - y^{\mathrm{T}} T_k y) = 0$,
5. $I + (\sigma - \mu)T \succeq 0$.

Recall that the shifted and inverted GLTR algorithm acts as Lanczos-CG until either the trust-region boundary is encountered or the matrix $H$ is detected not to be positive definite. Thus, if the algorithm switches to trust-region mode, then the solution to the trust-region problem must lie on the trust-region boundary. Problem 4.20 can be simplified to

$$\begin{aligned} &\min_{y \in \mathbb{R}^k} \frac{1}{2} y^{\mathrm{T}} (I - \mu T_k) y + \beta e_1 y \\ &\text{s.t. } ||y||_{T_k} = \delta, \end{aligned} \tag{4.27}$$

which has optimality conditions

1. $(I + (\sigma - \mu)T)y + \beta e_1 = 0$,

2. $\delta^2 - y^{\mathrm{T}} T_k y = 0$,

3. $I + (\sigma - \mu)T \succeq 0$.

Let $\nu = \sigma - \mu$. Then the first condition becomes $(I + \nu T)y + \beta e_1 = 0$, and the third condition becomes $I + \nu T \succeq 0$. The second condition remains unchanged. These are the optimality conditions for the problem

$$\begin{aligned} &\min_{y \in \mathbb{R}^k} \frac{1}{2} y^{\mathrm{T}} y + \beta e_1 y \\ &\text{s.t. } ||y||_{T_k} = \delta. \end{aligned} \tag{4.28}$$

This problem can easily be solved via the Moré-Sorensen algorithm, just as in the unshifted GLTR algorithm. However, one modification needs to be made that is worth mentioning. The Moré-Sorensen algorithm begins by creating an interval in which the optimal $\nu$ must lie. BY (4.15), a sufficient choice of the initial interval is

$$\frac{||\beta e_1||_{T_k^{-1}}}{\delta} - \lambda_1^{(k)} \leq \nu \leq \frac{||\beta e_1||_{T_k^{-1}}}{\delta} - \lambda_n^{(k)},$$

where $\lambda_1^{(k)}$, and $\lambda_n^{(k)}$ are the largest and smallest eigenvalues of the matrix pencil $(I, T_k)$, respectively. However, computing both $\lambda_1^{(k)}$ and $\lambda_n^{(k)}$ is, in general, more difficult than computing $\nu^{(k)}$, the optimal Lagrange multiplier. Instead,

$$\frac{||\beta e_1||_{T_k^{-1}}}{\delta} - u \leq \nu \leq \frac{||\beta e_1||_{T_k^{-1}}}{\delta} - l,$$

where $l \leq \lambda_n$ and $u \geq \lambda_1$ are constants found via the Gershgorin circle theorem.

Typically, in the Moré-Sorensen algorithm, in the case where the matrix which defines the trust region is the identity matrix, this interval is approximated by replacing the eigenvalues with the upper and lower bounds given via the Gershgorin circle theorem. However, unless $T_k$ is strictly diagonally dominant, there is no equally convenient theorem to find upper and lower bounds on the eigenvalues of the pencil $(I, T_k)$.

Consider the generalized eigenvalue problem $s = \lambda T_k s$. This is a banded symmetric positive definite generalized eigenvalue problem. Thus all the eigenvalues are positive. Next, consider the related problem $T_k s = \gamma s$. Thus, $\lambda_1 = 1/\gamma_n$, and $\lambda_n = 1/\gamma_1$, and Gershgorin circle bounds for the problem $T_k s = \gamma s$ can be used for the problem $s = \lambda T_k s$. Alternatively, a split Cholesky decomposition of the matrix $T_k$ can be used to transform the problem $s = \lambda T_k s$ to a problem of the form $C_k s = \lambda s$, where $C_k$ is a tridiagonal matrix. The LAPACK routine DPBSTF can be used to perform this transformation.

### 4.5.2 Solving in the Hard Case

The GLTR algorithm, as originally presented in [18], does not attempt to solve problem (4.1) correctly if it is an instance of the hard case. The SIGLTR algorithm, as presented thus far, suffers from the same issue, i.e., if $g \in \text{null}(H-\lambda_n M)^{\perp}$, then all subsequent Lanczos vectors $u_i$ will lie in the same subspace, and a solution of the form $x = (H - \lambda_n M)^{\dagger} + \tau u_n$ will never be found. Fortunately, there exists a simple correction to mitigate this issue. A block-Lanczos process can replace the standard Lanczos process used to construct the basis for solving the projected subproblem. Let $m \geq 1$ be the predetermined block size. Let $\tilde{V}_1 \in \mathbb{R}^{n\times m}$, where $\tilde{V}_1 e_1 = -g$, and $\tilde{V}_1 e_i$ for $i = 2, \ldots, m$ be Gaussian random vectors with variance $||g||^2_{(H+\mu M)^{-1}}$. Let $\tilde{U}_1 = (H + \mu M)^{-1}\tilde{V}_1$. With high probability, $\text{range}(\tilde{U}_1) \cap \text{null}(H - \lambda_n M) \neq \{0\}$. By utilizing the Gram-Schmidt biorthogonalization process, matrices $U_1, V_1 \in \mathbb{R}^{n\times m}$, and $B_1 \in \mathbb{R}^{m\times m}$ upper-triangular can be found such that $\tilde{U}_1 = U_1 B_1$ and $\tilde{V}_1 = V_1 B_1$. Note then that $U_1^{\mathrm{T}} g = B_1 e_1 \in \mathbb{R}^m$. The block-Lanczos process then proceeds like the standard Lanczos process, replacing the normalization step with a Gram-Schmidt biorthogonalization step.

In the Lanczos process with $m = 1$, breakdown occurs when $\beta_{j+1} = 0$. In this case, no other Lanczos vectors can be found without initializing a new vector $u_{j+1}$ that is $M$-orthogonal to all previous Lanczos vectors. In exact arithmetic, breakdown occurs when the Lanczos vectors computed thus far span the smallest $(H + \mu M)^{-1}M$-invariant subspace containing the vector $(H + \mu M)^{-1}g$. Fortunately, the GLTR and SIGLTR algorithms do not need to take this extra step, as the convergence criteria show that if a breakdown occurs, the algorithm has converged. Typically, convergence will occur far before breakdown. However, as this algorithm should also function in the low dimensional case, this must still be considered in any practical implementation.

In the block-Lanczos process, extra care must be taken. Breakdown occurs in this case when $\text{rank}(B_{j+1}) = r_{j+1} < m$. If $r_{j+1} > 0$, then an additional $r_{j+1}$ Lanczos vectors are obtained with which to search for a solution. Thus, one additional iteration is carried out with $U_{j+1} \in \mathbb{R}^{n\times r_{j+1}}$, $B_{j+1} \in \mathbb{R}^{r_{j+1}\times m}$, and $A_{j+1} \in \mathbb{R}^{r_{j+1}\times r_{j+1}}$. Subsequently, $B_{j+2} = 0$, so the convergence criteria indicate that convergence will occur in this final step. It is essential to carry out this final step in the low dimensional case when $m$ does not divide $n$. In the block case, the residual vector can be computed via the expression

$$r_k^{\mathrm{T}}(H + \mu M)^{-1} r_k = (\sigma_k - \mu)^2 ||B_{k+1} E_k y_k||_2^2, \tag{4.29}$$

where

$$E_k y_k = \begin{pmatrix} [y_k]_{k-m+1} \\ \vdots \\ [y_k]_k \end{pmatrix}$$

There is a trade-off for using the block version. The computation per iteration involving vectors of size $n$ is effectively multiplied by $m$ per iteration. Unfortunately, this does not hold for the number of iterations. As the block-size increases, the number of iterations only slightly decreases, with diminishing returns as $m \to n$. If $m = n$, the algorithm reduces to the Moré-Sorensen algorithm acting on a transformed trust-region problem. Thus, it is recommended to keep the block-size $m$ small.

### 4.5.3 The Full Algorithm

Here, the pseudocode for the full shifted-and-inverted GLTR algorithm is presented. This uses the block version without loss of generality to account for the hard case. In a typical implementation of a block-Lanczos process, the $k, k+1$ block of $T$, denoted $B_{k+1}$, is typically found by taking the QR decomposition of $\tilde{U}_{k+1} = HU_k - U_k A_k - U_{k-1} B_k^{\mathrm{T}}$ to get $U_{k+1} B_{k+1} = \tilde{U}_{k+1}$. As the block size is typically on the order of $m \leq 10$, and the dimension $n$ may be enormous, the Gram-Schmidt version of the QR decomposition is the best choice. Two modifications are made. First, due to the use of the matrix $H + \mu M$ over the Euclidean inner product, the biorthogonal Gram-Schmidt process is used to form $U_{k+1} B_{k+1} = \tilde{U}_{k+1}$ such that $U_{k+1}^{\mathrm{T}}(H + \mu M) U_{k+1} = I_m$. Additionally, the QR decomposition with column pivoting is used to detect better when a breakdown occurs. The biorthogonalization process with column pivoting is presented in Algorithm 4.1.

**Algorithm 4.1.** Biorthogonalization with Column Pivoting

Given $\tilde{U} \in \mathbb{R}^{n\times m}, \tilde{V} \in \mathbb{R}^{n\times m}$
Yields $U$, $V$ such that $V^{\mathrm{T}}U = I_m$, $M$ is an upper triangular matrix with permuted columns, and $P$ is a permutation matrix such that $UM = \tilde{U}P$, and $VM = \tilde{V}P$, and $r = \mathrm{rank}(\tilde{U})$
$U \leftarrow \tilde{U}$
$V \leftarrow \tilde{V}$
$B \leftarrow 0_{m\times m}$
$r \leftarrow 0$
ipiv $\leftarrow \begin{bmatrix} 1 & 2 & \cdots & m \end{bmatrix}$
**for** i = 1, . . . , m **do**
    $\mathrm{norms}_i \leftarrow (V_i^{\mathrm{T}}U_i)^{1/2}$
**end for**
**for** i=1, . . . , m **do**
    $k \leftarrow \max_{k\in\{i,\ldots,m\}} \mathrm{norms}_k$
    **if** $\mathrm{norms}_k < \sqrt{\varepsilon}$ **then** exit
    **end if**
    rank $\leftarrow$ rank + 1
    **if** $k \neq i$ **then**
        Swap $U_i$ and $U_k$
        Swap $V_i$ and $V_k$
        Swap $\mathrm{norms}_i$ and $\mathrm{norms}_k$
        Swap $\mathrm{ipiv}_i$ and $\mathrm{ipiv}_k$
    **end if**
    $U_i \leftarrow U_i/\mathrm{norms}_i$
    $V_i \leftarrow V_i/\mathrm{norms}_i$
    $B_{i,\mathrm{ipiv}_i} \leftarrow \mathrm{norms}_i$
    **for** j= i + 1, . . . , m **do**
        $B_{i,\mathrm{ipiv}_j} \leftarrow V_i^{\mathrm{T}}U_j$
        $V_j \leftarrow V_j - B_{i,\mathrm{ipiv}_j}V_i$
        $U_j \leftarrow U_j - B_{i,\mathrm{ipiv}_j}U_i$
        $\mathrm{norms}_j \leftarrow |\mathrm{norms}_j^2 - B_{i,\mathrm{ipiv}_j}^2|^{1/2}$
    **end for**
**end for**
$P \leftarrow$ Permutation matrix form of ipiv.

The full SIGLTR algorithm is presented in Algorithm 4.2.

**Algorithm 4.2.** Shifted and Inverted GLTR

Given $H, M \in \mathbb{R}^{n\times n}$, $B \succ 0$, $\mu \geq 0$ such that $H + \mu M \succ 0$, $g \in \mathbb{R}^n$, and $\delta > 0$.
Let $\varepsilon_1, \varepsilon_2 > 0$
Let $m \geq 1$ be the block-size.
$k \leftarrow 0$
$\tilde{V} \leftarrow \begin{bmatrix} g & v_2 & \cdots & v_m \end{bmatrix}$, where $v_2, \ldots, v_m$ are Gaussian random vectors.
$\tilde{U} \leftarrow (H + \mu M)^{-1}\tilde{V}$.
$(U, V, B_0) \leftarrow$ Biorthogonalize$(\tilde{U}, \tilde{V})$
$\beta \leftarrow B_0 e_1$
$x \leftarrow 0$
$(x^{\mathrm{T}}Mx) \leftarrow 0$
$(P^{\mathrm{T}}Mx) \leftarrow 0_m$
$(P^{\mathrm{T}}MP) \leftarrow 0_{m\times m}$
$A_0 \leftarrow 0_{m\times m}$
$B_0 \leftarrow 0_{m\times m}$
$D \leftarrow I_m$
$D_{\text{prev}} \leftarrow 0_{m\times m}$
$L \leftarrow 0_{m\times m}$
$z \leftarrow 0_m$
$V_{\text{prev}} \leftarrow 0$
mode $\leftarrow$ CG
TR Solved $\leftarrow$ $false$
breakdown $\leftarrow$ $false$
**while** not converged **do**
    $k \leftarrow k + 1$
    **if** $k > 1$ **then**
        Store $B_k$ in $T_{k,k-1}$
    **end if**
    $W \leftarrow MU - V_{\text{prev}}B^{\mathrm{T}}$
    $A_k \leftarrow U^{\mathrm{T}}W$
    Store $A_k$ in $T_{k,k}$
    **if** mode = CG **then**
        $D_{\text{prev}} \leftarrow D$
        $L \leftarrow -\mu B_k D_{\text{prev}}^{-1}$
        $D \leftarrow (I_m - \mu A_k) - LD_{\text{prev}}L^{\mathrm{T}}$
        **if** $D \succ 0$ **then**
            $P \leftarrow (U + \mu P B_k^{\mathrm{T}})D^{-1}$
            $(P^{\mathrm{T}}Mx) \leftarrow D^{-1}(B_k D_{\text{prev}}^{-1} z + \mu B_k((P^{\mathrm{T}}Mx) + (P^{\mathrm{T}}MP)z))$
            **if** $k = 1$ **then**
                $z \leftarrow -\beta$
            **else**
                $z \leftarrow -Lz$
            **end if**
            $||r||^2_{(H+\mu M)^{-1}} = z^{\mathrm{T}}z$
            **if** $||r||^2_{(H+\mu M)^{-1}} \leq \varepsilon_1$ **then**
                converged $\leftarrow$ true
            **else**
                $x \leftarrow x + Pz$
                $(P^{\mathrm{T}}MP) \leftarrow D^{-1}(A_k + 2\mu B_k D_{\text{prev}}^{-1} B_k^{\mathrm{T}} + \mu^2 M(P^{\mathrm{T}}MP)B_k^{\mathrm{T}})D^{-1}$
                $(x^{\mathrm{T}}Mx) \leftarrow (x^{\mathrm{T}}Mx) + 2z^{\mathrm{T}}(P^{\mathrm{T}}Mx) + z^{\mathrm{T}}(P^{\mathrm{T}}MP)z$
                **if** $\sqrt{(x^{\mathrm{T}}Mx)} > (1 + \varepsilon_2)\delta$ **then**
                    mode $\leftarrow$ TR
                **end if**
            **end if**

**else** ▷ If $D$ is indefinite
    mode $\leftarrow$ TR
**end if**
**end if**
**if** TR Solved and not converged **then**
    $||r||^2_{(H+\mu M)^{-1}} = (\sigma - \mu)||B_k E_k y||^2$
    **if** $||r||^2_{(H+\mu M)^{-1}} \leq \varepsilon_1$ **then**
        converged $\leftarrow$ true
    **end if**
**end if**
**if** not converged and mode $\neq CG$ **then**
    Solve (4.20) for $y$ and $\sigma$.
    TR Solved $\leftarrow$ true
**end if**
**if** breakdown **then**
    exit
**end if**
**if** converged **then**
    exit
**end if**
$W \leftarrow W - VA_k$
$V_{\text{prev}} \leftarrow V$
$V \leftarrow W$
$U \leftarrow (H + \mu M)^{-1} V$
$(U, V, B_{k+1}, \text{rank}) \leftarrow \text{BOCP}(U, V)$
**if** rank $< m$ **then**
    breakdown $\leftarrow$ true
    **if** rank $= 0$ **then**
        exit
    **end if**
**end if**
Optional: Orthogonalize $U$ and $V$ against previous Lanczos vectors.
**end while**
**if** mode = TR **then**
Build $x$ from $y$ and regenerated or stored vectors $U$
**end if**

### 4.5.4 Choice of Shift

One question that remains to be answered is how to choose the shift $\mu$. The ideal choice would be to choose $\mu = \sigma$. Thus, the goal is to approximate $\sigma$ as close as possible. Recall that

$$\max\{0, -\lambda_n, \frac{||g||_{M^{-1}}}{\delta} - \lambda_1\} \leq \sigma \leq \max\{0, \frac{||g||_{M^{-1}}}{\delta} - \lambda_n\}.$$

The choice of $\mu$ should attempt to remain within these bounds. Consider the optimality condition $(H + \sigma M)x + g = 0$. If $x$ is the optimal solution and is a constrained minimizer, then

$$\sigma(x) = \frac{-g^{\mathrm{T}}x - x^{\mathrm{T}}Hx}{x^{\mathrm{T}}Mx}.$$

This expression can be used to construct a guess $\sigma$ from any approximate solution on the boundary. The only vector available before any computation is $g$, so suppose an initial guess of $\sigma$ of the form

$$\sigma(\frac{-\delta}{||g||_M}g) = \frac{g^{\mathrm{T}}g}{\delta||g||_M} - \frac{g^{\mathrm{T}}Hg}{g^{\mathrm{T}}Mg}.$$

is chosen. Notice that the second term is simply the Rayleigh quotient of the vector $g$. Since $\max\{-\lambda_n, ||g||_{M^{-1}}/\delta - \lambda_1\} \leq \sigma \leq ||g||_{M^{-1}}/\delta - \lambda_n$, $-g^{\mathrm{T}}Hg/g^{\mathrm{T}}Mg$ can be replaced with $-\lambda_n$ to ensure that $\mu$ yields a positive definite matrix. Thus,

$$\mu = \frac{g^{\mathrm{T}}g}{\delta||g||_M} - \lambda_n \tag{4.30}$$

is a reliable choice of $\mu$.

Note that the Cauchy-Schwartz inequality implies $g^{\mathrm{T}}g/||g||_M \leq ||g||_{M^{-1}}$, so $-\lambda_n \leq \mu \leq ||g||_{M^{-1}}/\delta - \lambda_n$. Of course, the complexity of finding $\lambda_n$ generally exceeds that of finding $\sigma$, so an approximation should be used. Approximations can be found via Gerschgorin circles, or from a few steps of any iterative eigenvalue algorithm.

### 4.5.5 Convergence Properties

Before analyzing the rate of convergence of the shifted and inverted GLTR algorithm, it will help to prove that the sequence of Lagrange multipliers $\{\sigma_k\}_{k=1}^{k_{max}}$ generated satisfies $\sigma_1 \leq \sigma_2 \leq \ldots \leq \sigma_{k_{max}} \leq \sigma^*$. First, consider the case $||x^*||_M < \delta$, so that $\sigma_k = \sigma^* = 0$ for all iterations $k$. The following lemma demonstrates the relationship between the Krylov subspace used in the shifted and inverted GLTR algorithm and the Krylov subspace used in the preconditioned conjugate-gradient algorithm.

**Lemma 4.5.1.** *Let $H$, $M \in \mathbb{R}^{n\times n}$ be symmetric matrices, with $M$ additionally positive definite, and let $\mu > 0$. If $\mu = 0$ and $\sigma^* = 0$, the shifted and inverted GLTR algorithm converges in one iteration. Let $C = H + \mu M$. Let*

$$\mathcal{K}_k(C^{-1}M, C^{-1}g) = \mathrm{span}\{C^{-1}g, C^{-1}MC^{-1}g, \ldots, (C^{-1}M)^{k-1}C^{-1}g\}$$

*be the k-th Krylov subspace searched in the shifted-and-inverted GLTR method. Then*

$$\mathcal{K}_k(C^{-1}M, C^{-1}g) = \mathcal{K}_k(C^{-1}H, C^{-1}g)$$

*Proof.* The lemma immediately holds for $k = 1$. Suppose that lemma (4.5.1) holds for some $k > 1$. Then

$$(C^{-1}H)^k C^{-1}g = (C^{-1}H)(C^{-1}H)^{k-1}C^{-1}g = C^{-1}Hv$$

for some $v \in \mathcal{K}_k(C^{-1}M, C^{-1}g) = \mathcal{K}_k(C^{-1}H, C^{-1}g)$. Now,

$$C^{-1}Hv = C^{-1}Cv - \mu C^{-1}Mv = v - \mu C^{-1}Mv \in \mathcal{K}_{k+1}(C^{-1}M, C^{-1}g),$$

so $\mathcal{K}_{k+1}(C^{-1}H, C^{-1}g) \subseteq \mathcal{K}_{k+1}(C^{-1}M, C^{-1}g)$. Similarly,

$$(C^{-1}M)^k C^{-1}g = (C^{-1}M)(C^{-1}H)^{k-1}C^{-1}g = C^{-1}Mv$$

for some $v \in \mathcal{K}_k(C^{-1}M, C^{-1}g) = \mathcal{K}_k(C^{-1}H, C^{-1}g)$. Now,

$$C^{-1}Mv = \frac{1}{\mu}C^{-1}Cv - \frac{1}{\mu}C^{-1}Hv = \frac{1}{\mu}v - \frac{1}{\mu}C^{-1}Hv \in \mathcal{K}_{k+1}(C^{-1}H, C^{-1}g),$$

so $\mathcal{K}_{k+1}(C^{-1}M, C^{-1}g) \subseteq \mathcal{K}_{k+1}(C^{-1}H, C^{-1}g)$. □

**Corollary 4.5.1.1.** *It holds that*

$$\mathcal{K}_k(C^{-1}M, C^{-1}g) = \mathcal{K}_k(C^{-1}(H + \sigma M), C^{-1}g)$$

*for any $\sigma \neq \mu$.*

**Lemma 4.5.2.** *Let $H$ and $M$ be symmetric positive definite matrices, and let $\mu \geq 0$. Let $C = H + \mu M$. Let*

$$\mathcal{K}_k(C^{-1}M, C^{-1}g) = \text{span}\{C^{-1}g, C^{-1}MC^{-1}g, \ldots, (C^{-1}M)^{j-1}C^{-1}g\}$$

*be the k-th Krylov subspace searched by the shifted-and-inverted GLTR method, and*

$$x_k = \underset{x \in \mathcal{K}_k(C^{-1}M, C^{-1}g)}{\text{argmin}} \frac{1}{2}x^{\mathrm{T}}Hx + g^{\mathrm{T}}x.$$

*If* $1 \leq k \leq l \leq n$, *then* $||x_k||_C \leq ||x_l||_C$.

*Proof.* The lemma follows from Lemma 4.5.1 and the well-known fact that the conjugate-gradient iterates increase in the norm induced by the preconditioner as the algorithm proceeds. □

From the construction of phase one of the SIGLTR algorithm, the following result holds:

**Lemma 4.5.3.** *Let* $H, M$ *be symmetric positive definite matrices, and let* $\mu \geq 0$. *Let* $C = H + \mu M$. *Let*

$$\mathcal{K}_k(C^{-1}M, C^{-1}g) = \text{span}\{C^{-1}g, C^{-1}MC^{-1}g, \ldots, (C^{-1}M)^{j-1}C^{-1}g\}$$

*be the* $k$*-th Krylov subspace searched by the shifted-and-inverted GLTR method, and*

$$x_k = \underset{x \in \mathcal{K}_k(C^{-1}M, C^{-1}g)}{\text{argmin}} \frac{1}{2}x^{\mathrm{T}}Hx + g^{\mathrm{T}}x.$$

*If* $1 \leq i \leq j \leq n$, *then* $||x_i||_M \leq ||x_j||_M$.

To simplify notation, let $\mathcal{S}_k = \mathcal{K}_k(C^{-1}M, C^{-1}g)$. Let $Z_k$ be any $C$-orthonormal basis for $\mathcal{S}_k$.

**Lemma 4.5.4.** *Let* $H$ *be symmetric,* $M$ *be symmetric positive definite,* $\mu \geq 0$, *and* $C = H + \mu M$ *positive definite. Let* $Z_kHZ_k + \sigma_i Z_kMZ_k$ *be positive definite for* $i \in \{1, 2\}$. *Let*

$$x_k^{(i)} = \underset{x \in \mathcal{S}_k}{\text{argmin}} \frac{1}{2}x^{\mathrm{T}}(H + \sigma_i M)x + g^{\mathrm{T}}x.$$

*Then*

$$\sigma_2 \leq \sigma_1 \iff ||x_k^{(2)}||_C \geq ||x_k^{(1)}||_C.$$

*Proof.* For each $i$, $x_k^{(i)}$ is given by $Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k y_k^{(i)} = -Z_k^{\mathrm{T}}g$, and $x_k = Z_k y_k$. As $H + \sigma_i M$ is nonsingular, $Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k$ is as well. Therefore, $y_k^{(i)} = -(Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k)^{-1}Z_k^{\mathrm{T}}g$, and

$$x_k^{(i)} = -Z_k(Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k)^{-1}Z_k^{\mathrm{T}}g.$$

Thus,

$$||x_k^{(i)}||_C^2 = g^{\mathrm{T}}Z_k(Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k)^{-2}Z_k^{\mathrm{T}}g$$

The result follows. □

While the above result does have some useful implications, the trust-region problem is more concerned with $||x||_M$ than $||x||_C$. Thus, the following result is needed.

**Lemma 4.5.5.** *Let $H$ be symmetric and $M$ be positive definite. Let $\mu \geq 0$ and $C = H + \mu M$ be positive definite. Let $Z_k H Z_k + \sigma_i Z_k M Z_k$ be positive definite for $i \in \{1, 2\}$. Let*

$$x_k^{(i)} = \operatorname*{argmin}_{x \in \mathcal{S}_k} \frac{1}{2} x^{\mathrm{T}}(H + \sigma_i M)x + g^{\mathrm{T}}x.$$

*Then*

$$\sigma_2 \leq \sigma_1 \iff ||x_k^{(2)}||_M \geq ||x_k^{(1)}||_M.$$

*Proof.* For each $i$, $x_k^{(i)}$ is given by $Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k y_k^{(i)} = -Z_k^{\mathrm{T}} g$, and $x_k = Z_k y_k$. As $H + \sigma_i M$ is non-singular, $Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k$ is as well. Thus $y_k^{(i)} = -(Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k)^{-1} Z_k^{\mathrm{T}} g$, and

$$x_k^{(i)} = -Z_k(Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k)^{-1} Z_k^{\mathrm{T}} g.$$

Thus

$$||x_k^{(i)}||_M^2 = g^{\mathrm{T}} Z_k (Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k)^{-1} Z_k^{\mathrm{T}} M Z_k (Z_k^{\mathrm{T}}(H + \sigma_i M)Z_k)^{-1} Z_k^{\mathrm{T}} g$$

To simplify the notation, let $\bar{g} = Z_k^{\mathrm{T}} g$, $\bar{H} = Z_k^{\mathrm{T}} H Z_k$, and $\bar{M} = Z_k^{\mathrm{T}} M Z_k$. Then

$$||x_k^{(i)}||_M^2 = \bar{g}^{\mathrm{T}}(\bar{H} + \sigma_i \bar{M})^{-1} \bar{M} (\bar{H} + \sigma_i \bar{M})^{-1} \bar{g}.$$

Then

$$||x_k^{(2)}||_M^2 - ||x_k^{(i)}||_M^2 = \bar{g}^{\mathrm{T}}\big((\bar{H} + \sigma_2 \bar{M})^{-1} \bar{M} (\bar{H} + \sigma_2 \bar{M})^{-1} - (\bar{H} + \sigma_1 \bar{M})^{-1} \bar{M} (\bar{H} + \sigma_1 \bar{M})^{-1}\big) \bar{g}.$$

Recall that for two nonsingular matrices $X, Y \in \mathbb{R}^{n \times n}$, $X - Y \succeq 0$ if and only if $Y^{-1} - X^{-1} \succeq 0$. With this in mind, consider the matrix

$$\begin{aligned}(\bar{H} + \sigma_1 \bar{M})\bar{M}^{-1}(\bar{H} + \sigma_1 \bar{M}) - (\bar{H} + \sigma_2 \bar{M})\bar{M}^{-1}(\bar{H} + \sigma_2 \bar{M}) \\ = 2(\sigma_1 - \sigma_2)(\bar{H} + \sigma_2 \bar{M}) + (\sigma_1 - \sigma_2)^2 \bar{M} \succeq 0.\end{aligned}$$

The result follows. □

It can now be shown that the sequence of Lagrange multipliers is non-decreasing.

**Theorem 4.5.6.** *Let $H$ be symmetric, $M$ symmetric positive definite, and $\mu \geq 0$ such that $C = H + \mu M \succ$*

0. *For $k \in \{1, \ldots, n\}$, let $x_k$ be the solution to*

$$\min_{x \in \mathcal{S}_k} \frac{1}{2} x^{\mathrm{T}} H x + g^{\mathrm{T}} x$$

$$s.t. \; ||x||_M \leq \delta$$

*with corresponding Lagrange multipliers $\sigma_k$. If $1 \leq i \leq j \leq n$, then*

$$\sigma_i \leq \sigma_j.$$

*Proof.* For each $\mathcal{S}_k$, let $Z_k$ be a $C$-orthonormal basis for $\mathcal{S}_k$, and let $x_k = Z_k y_k$ for all $k$. Then, by Theorem 4.1.1, $x_k$ solves the $k$-th trust-region subproblem if and only if

1. $Z_k^{\mathrm{T}}(H + \sigma_k M) Z_k y_k = -Z_k^{\mathrm{T}} g$,
2. $\sigma_k \geq 0$,
3. $\delta^2 - y_k^{\mathrm{T}} Z_k^{\mathrm{T}} M Z_k y_k \geq 0$,
4. $\sigma_j(\delta^2 - y_k^{\mathrm{T}} Z_k^{\mathrm{T}} M Z_k y_k) = 0$,
5. $Z_k^{\mathrm{T}}(H + \sigma_k M) Z_k \succeq 0$.

Recall that $x_k$ lies strictly within the trust region only if $\sigma_k = 0$; otherwise, it lies on the boundary. Thus, if $i < j$ and $\sigma_j = 0$, $||x_i||_M \leq ||x_j||_M < \delta$, so $\sigma_i = 0$. If $\sigma_i = 0$ and $\sigma_j \geq 0$, there is nothing to show. Thus, assume that $\sigma_i > 0$ and $\sigma_j > 0$, so that $||y_i||_{Z_i^{\mathrm{T}} M Z_i} = ||x_i||_M = ||y_j||_{Z_j^{\mathrm{T}} M Z_j} = ||x_j||_M = \delta$. First, assume that $Z_i^{\mathrm{T}}(H + \sigma_i M) Z_i$ is singular, and that $\sigma_j < \sigma_i$. Then there exists some vector $w \in \mathcal{K}_i \subset \mathcal{K}_j$ such that $w^{\mathrm{T}}(H + \sigma_j M) w < 0$. Thus, $Z_j^{\mathrm{T}}(H + \sigma_j M) Z_j$ is not positive semidefinite, so $\sigma_j$ is not the Lagrange multiplier of the $j$-th trust-region subproblem. Thus, $\sigma_i \leq \sigma_j$.

Now, assume that both $Z_i^{\mathrm{T}}(H + \sigma_i M) Z_i \succ 0$ and $Z_j^{\mathrm{T}}(H + \sigma_j M) Z_j \succ 0$. By Corollary 4.5.1.1, $x_i$ is a solution to the unconstrained minimization problem

$$\min_{x \in \mathcal{S}_i} \frac{1}{2} x^{\mathrm{T}}(H + \sigma_i M) x + g^{\mathrm{T}} x.$$

For the sake of contradiction, assume that $\sigma_i > \sigma_j$. This implies that $Z_j^{\mathrm{T}}(H + \sigma_i M) Z_j \succ 0$. Let $\tilde{x_j}$ be the solution to

$$\min_{x \in \mathcal{S}_j} \frac{1}{2} x^{\mathrm{T}}(H + \sigma_i M) x + g^{\mathrm{T}} x.$$

By Lemma 4.5.3, $||\tilde{x_j}||_M \geq ||x_i||_M = ||x_j||_M = \delta$. By Lemma 4.5.5, $\sigma_i \leq \sigma_j$, which is a contradiction. Finally, assume that $Z_j^{\mathrm{T}}(H + \sigma_j M)Z_j$ is singular, so that $Z_j^{\mathrm{T}}(H + (\sigma_j + \varepsilon)M)Z_j \succ 0$. Then $Z_i^{\mathrm{T}}(H + (\sigma_j + \varepsilon)M)Z_i \succ 0$. Let $\widehat{x}_i$ and $\widehat{x}_j$ solve

$$\min_{x \in \mathcal{K}_i} \frac{1}{2} x^{\mathrm{T}}(H + (\sigma_j + \varepsilon)M)x + g^{\mathrm{T}}x \quad \text{and} \quad \min_{x \in \mathcal{K}_j} \frac{1}{2} x^{\mathrm{T}}(H + (\sigma_j + \varepsilon)M)x + g^{\mathrm{T}}x,$$

respectively. Then $||\widehat{x}_i||_M \leq ||\widehat{x}_j||_M \leq \delta$. As $||x_i||_M = \delta$, $\sigma_i \leq \sigma_j + \varepsilon$. As $\varepsilon$ can be made arbitrarily small, $\sigma_i \leq \sigma_j$. The result follows. □

Theorem 4.4.1 shows that the standard GLTR algorithm converges towards the optimal solution $x^*$ in a manner identical to that of the preconditioned conjugate-gradient algorithm for solving the system $(H + \sigma^* M)x + g = 0$ when $\sigma^*$ is known, and $M$ is used as a preconditioner, assuming that the trust-region problem being solved is not an instance of the hard-case. It stands to reason that a similar result could be shown for SIGLTR, where the convergence is instead identical to solving the same system using $H + \mu M$ as a preconditioner. The proof presented in [23] is closely followed with some technical details changed to show this similar result.

Without loss of generality, assume that the trust-region constraint matrix $M = I$. Adachi et al. prove in [1] that the trust-region subproblem is equivalent to solving a $2n \times 2n$ generalized eigenvector problem of the matrix

$$N = \begin{pmatrix} -H & \frac{1}{\delta^2} g g^{\mathrm{T}} \\ I & -H \end{pmatrix} \in \mathbb{R}^{2n \times 2n} \tag{4.31}$$

Let $\mu_1, \ldots, \mu_{2n}$ be the eigenvalues of $N$ ordered in decreasing order of the real parts, i.e.,

$$Re(\mu_1) \geq \cdots \geq Re(\mu_{2n}).$$

Then the following intuitive theorem holds:

**Theorem 4.5.7.** *Let $(x^*, \sigma^*)$ be the optimal primal-dual solution to the trust-region problem defined by $H$, $g$, and $\delta$, with $||x^*|| = \delta$. Then the leftmost eigenvalue $\mu_1$ of $N$ is real and simple, and $\mu_1 = \sigma^*$. Let $y = (y_1, y_2) \in \mathbb{R}^{2n}$ be the eigenvector of $N$ corresponding to $\mu_1$, and suppose that $g^{\mathrm{T}} y_2 \neq 0$. Then the unique solution to the trust-region problem is*

$$x^* = -\frac{\delta^2}{g^{\mathrm{T}} y_2} y_1.$$

*Remark.* Adachi et al. also prove that $g^{\mathrm{T}} y_2 = 0$ corresponds to the hard case. By assumption, the problem is not an instance of the hard case, which guarantees that $g^{\mathrm{T}} y_2 \neq 0$.

Let $\{(x_k, \sigma_k)\}$ be the iterates generated by the SIGLTR algorithm with shift $\mu$ such that $C = H + \mu I \succ 0$. Let $k^*$ be the final iteration index. It has already been shown in Theorem 4.5.6 that $0 \leq \sigma_1 \leq \ldots \leq \sigma_{k^*} = \sigma^*$. Assume that at iteration $K$, phase one ends and phase two begins, so that $||x_k|| = \delta$ and $\sigma_k > 0$ for each $k > K$. Let $U_k \in \mathbb{R}^{n \times k}$ be the $C$-orthogonal matrix consisting of the first $k$ Lanczos vectors. Use $\mathcal{S}_k$ to denote the Krylov subspace $\mathcal{K}_k(C^{-1}H, C^{-1}g)$. Then $\mathrm{range}(U_k) = \mathcal{S}_k$. Let

$$\tilde{U}_k = \begin{pmatrix} U_k & \\ & U_k \end{pmatrix},$$

and

$$\tilde{\mathcal{S}}_k = \mathrm{range}(\tilde{U}_k) = \mathcal{S}_k \oplus \mathcal{S}_k \subset \mathbb{R}^{2n}.$$

Let $D_k = I - \mu T_k$. Let

$$N_k = \tilde{U}_k^{\mathrm{T}} N \tilde{U}_k = \begin{pmatrix} -D_k & \frac{\beta^2}{\delta^2} e_1 e_1^{\mathrm{T}} \\ T_k & -D_k \end{pmatrix},$$

and

$$\tilde{T}_k = \begin{pmatrix} T_k & \\ & T_k \end{pmatrix}.$$

Then the projected eigenvalue problem is, at each iteration $k$,

$$N_k \begin{pmatrix} z_1^{(k)} \\ z_2^{(k)} \end{pmatrix} = \mu^{(k)} \tilde{T}_k \begin{pmatrix} z_1^{(k)} \\ z_2^{(k)} \end{pmatrix}.$$

By Theorem 4.5.7 and the construction of the SIGLTR algorithm, $\mu_1^{(k)} = \sigma_k$, and $\mu_1^{(k)}$ is real and simple. Let $z^{(k)} = \begin{pmatrix} z_1^{(k)} \\ z_2^{(k)} \end{pmatrix}$ be the eigenvector of $N_k$ associated with $\mu_1^{(k)}$, scaled so that $(z^{(k)})^{\mathrm{T}} \tilde{T}_k z^{(k)} = 1$. Note that $T_k$ is symmetric positive definite (as $T_k = U_k^{\mathrm{T}} U_k$), and therefore $\tilde{T}_k$ is as well. Then

$$y^{(k)} = \tilde{U}_k \begin{pmatrix} z_1^{(k)} \\ z_2^{(k)} \end{pmatrix} = \begin{pmatrix} U_k z_1^{(k)} \\ U_k z_2^{(k)} \end{pmatrix}$$

is the Ritz vector of $N$ associated with the rightmost eigenvalue $\mu_1 = \sigma^*$, and has unit length.

Recall that a left eigenvector of the pencil $(N_k, \tilde{T}_k)$ is a right eigenvector of the pencil $(N_k^{\mathrm{T}}, \tilde{T}_k^{\mathrm{T}}) = (N_k^{\mathrm{T}}, \tilde{T}_k)$. With this in mind, it is trivial to see that the left eigenvector of $(N_k, \tilde{T}_k)$ is

$$\begin{pmatrix} z_2^{(k)} \\ z_1^{(k)} \end{pmatrix}.$$

Alternatively, note that the generalized eigenvector problem $N_k z^{(k)} = \mu_k \tilde{T}_k z^{(k)}$ is equivalent to the standard eigenvector problem $\tilde{T}_k^{-1} N_k z^{(k)} = \mu^{(k)} z^{(k)}$, which has a left eigenvector

$$\begin{pmatrix} T_k z_2^{(k)} \\ T_k z_1^{(k)} \end{pmatrix}$$

associated with $\mu_1^{(k)}$. From the structure of $N_k$, it holds that

$$z_2^{(k)} = (D_k + \sigma_k T_k)^{-1} T_k z_1^{(k)} = (I + (\sigma_k - \mu) T_k)^{-1} T_k z_1^{(k)}.$$

Before proceeding, the following lesser-known definition of the spectral condition number of an eigenvalue is needed.

**Definition 4.5.1.** Let $\lambda$ be an eigenvalue of the matrix pencil $(M, N)$, where $N$ is symmetric positive-definite, and let $u$ and $v$ be the corresponding left and right eigenvectors. Then the spectral condition number of $\lambda$ is

$$s(\lambda) = \frac{||u||_N ||v||_N}{|v^{\mathrm{T}} N u|}.$$

Thus, the spectral condition number of $\mu_1^{(k)}$ is

$$s(\mu_1^{(k)}) = \frac{1}{|(z^{(k)})^{\mathrm{T}} \tilde{T}_k z^{(k)}|} = \frac{1}{2(z_1^{(k)})^{\mathrm{T}} T_k (D_k + \sigma_k T_k)^{-1} T_k z_1^{(k)}}. \tag{4.32}$$

Similarly, the spectral condition number of $\sigma^*$, the leftmost eigenvalue of $N$, is

$$s(\sigma^*) = \frac{1}{2 y_1^{\mathrm{T}} (H + \sigma^* I)^{-1} y_1}. \tag{4.33}$$

By Theorem 4.5.7, the unique solution to (4.20) is

$$h_k = -\frac{\delta^2}{\beta e_1^{\mathrm{T}} z_2^{(k)}} z_1^{(k)},$$

giving

$$x_k = U_k h_k = -\frac{\delta^2}{\beta e_1^{\mathrm{T}} z_2^{(k)}} y_1^{(k)}.$$

Let $\angle(u, V)$ denote the acute angle between a vector $u$ and a subspace $V$, so that

$$\sin\angle(u, V) = \frac{||(I - \pi)u||}{||u||},$$

where $\pi$ is the orthogonal projector onto $V$. This notion is generalized to the case where the finite-dimensional space in which $V$ is embedded uses a different inner product than the Euclidean inner product, as is the case here. Let $M$ be the positive-definite matrix that induces the inner product. Denote the $M$-angle between $u$ and $V$ by $\angle(u, V)_M$, so that

$$\sin\angle(u, V)_M = \frac{||(I - \pi)u||_M}{||u||_M},$$

where $\pi$ is now a $M$-orthogonal projector onto $V$. Let

$$\tilde{C} = \begin{pmatrix} C & \\ & C \end{pmatrix} = \begin{pmatrix} H + \mu I & \\ & H + \mu I \end{pmatrix}.$$

Let $\tilde{\pi}_k = \tilde{U}_k \tilde{U}_k^{\mathrm{T}} \tilde{C}$ be the $\tilde{C}$-orthogonal projector onto $\tilde{\mathcal{S}}_k$, and $\pi_k = U_k U_k^{\mathrm{T}} C$ be the $C$-orthogonal projector onto $\mathcal{S}_k$. At iteration $k$,

$$\sigma^* - \sigma_k = \mu_1 - \mu_1^{(k)}.$$

The following lemma from [22] can be applied.

**Lemma 4.5.8.** *Let $\mu_1^{(k)} = \sigma_k$ and $\mu_1 = \sigma^*$ be the rightmost eigenvalues of $N_k$ and $N$, respectively. Suppose that $||x^*|| = ||x_k|| = \delta$. Then for $\sin\angle(y, \tilde{S}_k)_{\tilde{C}}$ sufficiently small, it holds that*

$$\sigma^* - \sigma_k \leq s(\sigma_k)\tilde{\gamma}_k \sin\angle(y, \tilde{S}_k)_{\tilde{C}} + \mathcal{O}(\sin^2\angle(y, \tilde{S}_k)_{\tilde{C}}),$$

*where $s(\sigma_k)$ is defined by* (4.32)*, and $\tilde{\gamma}_k = ||\tilde{\pi}_k^{\mathrm{T}} N(I - \tilde{\pi}_k)||_{\tilde{C}}$*

From (4.32) and the definition of $y_k$,

$$s(\sigma_k) = \frac{1}{2|(y_2^{(k)})^{\mathrm{T}} y_1^{(k)}|},$$

which converges to $s(\sigma^*)$ defined by (4.33) as $y^{(k)} \to y$. Furthermore, $\tilde{\gamma}_k \leq ||N||_{\tilde{C}}$. Thus, in order to analyze how fast $\sigma_k$ converges to $\sigma^*$, it suffices to analyze how quickly $\sin \angle(y, \tilde{S}_k)_{\tilde{C}}$ decreases as $k \to k^*$. Now, notice that the definitions of $s(\sigma_k)$ and $s(\sigma^*)$ do not depend on the scaling of $(y_1^{(k)}, y_2^{(k)})$ and $(y_1, y_2)$. Let $y$ be rescaled so that $||y||_{\tilde{C}} = 1$. Then,

$$\sin^2 \angle(y, \tilde{S}_k)_{\tilde{C}} = \|(I - \tilde{\pi}_k) \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \|_{\tilde{C}}^2 = \|(I - \pi_k) y_1\|_{H+\mu I}^2 + \|(I - \pi_k) y_2\|_{H+\mu I}^2.$$

Before analyzing this expression, the relationship between the eigenvalues of $(H, C)$ and $(H + \sigma^* I, C)$ must be made clear.

**Lemma 4.5.9.** *Suppose the matrix pencil $(H, C)$ has eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$. Denote the eigenvalues of $(H + \sigma^* I, C)$ as $\tilde{\lambda}_1 \geq \ldots \geq \tilde{\lambda}_n$. Let $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ and $\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n)$. If $\mu = \sigma^*$, then $\tilde{\Lambda} = I$. If $\mu > \sigma^*$, then $\tilde{\lambda}_i = \frac{\sigma^*}{\mu} + \frac{\mu - \sigma^*}{\mu} \lambda_i$, and if $\mu < \sigma^*$, $\tilde{\lambda}_i = \frac{\sigma^*}{\mu} + \frac{\mu - \sigma^*}{\mu} \lambda_{n+1-i}$.*

*Proof.* Let $Q \in \mathbb{R}^{n \times n}$ such that $Q^{\mathrm{T}} C Q = I$ and $Q^{\mathrm{T}} H Q = \Lambda$, the diagonal matrix consisting of eigenvalues $\lambda_1, \ldots, \lambda_n$. Let $S = CQ$, so that $H = S \Lambda S^{\mathrm{T}}$, $S^{\mathrm{T}} Q = I$, and $S^{\mathrm{T}} C^{-1} S = I$. Then

$$H + \sigma^* I = \frac{\sigma^*}{\mu}(H + \mu I) + (1 - \frac{\sigma^*}{\mu})H = \frac{\sigma^*}{\mu} C + (1 - \frac{\sigma^*}{\mu})H,$$

thus

$$Q^{\mathrm{T}}(H + \sigma^* I)Q = \frac{\sigma^*}{\mu} I + \frac{\mu - \sigma^*}{\mu} \Lambda,$$

and

$$H + \sigma^* I = \frac{\sigma^*}{\mu} S S^{\mathrm{T}} + \frac{\mu - \sigma^*}{\mu} S \Lambda S^{\mathrm{T}}.$$

Two cases must be considered: $\mu > \sigma^*$, and $\mu < \sigma^*$. $\mu = \sigma^*$ has $n$ unit eigenvalues. First, consider $\mu > \sigma^*$. Then, for $i = 1, \ldots, n$,

$$\tilde{\lambda}_i = \frac{\sigma^*}{\mu} + \frac{\mu - \sigma^*}{\mu} \lambda_i.$$

On the other hand, if $\mu < \sigma^*$, then

$$\tilde{\lambda}_i = \frac{\sigma^*}{\mu} + \frac{\mu - \sigma^*}{\mu} \lambda_{n+1-i}.$$

□

First, $\|(I - \pi_k) y_1\|_C$ is analyzed. Let $P_k$ denote the set of polynomials of degree no larger than

$k+1$. Consider the following lemma:

**Lemma 4.5.10.** *The distance $\|(I-\pi_k)x^*\|_C$ between $x^*$ and the Krylov subspace $\mathcal{S}_k$ satisfies*

$$\|(I-\pi_k)x^*\|_C = \min_{p_k\in P_k, p_k(0)=1} \|p_k\big(C^{-1}(H+\sigma^* I)\big)x^*\|_C,$$

*and*

$$\|(I-\pi_k)x^*\|_C \le ||x_\star||_C\, \varepsilon_1^{(k)},$$

*where*

$$\varepsilon_1^{(k)} = \min_{p\in P_k, p(0)=1} \max_{1\le i\le n} ||p(\frac{\sigma^*}{\mu} + \frac{\mu-\sigma^*}{\mu}\lambda_i)||,$$

*with $\lambda_1 \ge \ldots \ge \lambda_n$ the generalized eigenvalues of the pencil $(H, C)$. Furthermore,*

$$\varepsilon_1^{(k)} \le 2\left(\frac{\sqrt{\kappa_C(H+\sigma^* I)}-1}{\sqrt{\kappa_C(H+\sigma^* I)}+1}\right)^{k+1}.$$

*where*

$$\kappa_C(H+\sigma^* I) = \frac{\tilde{\lambda}_1}{\tilde{\lambda}_n}$$

*is the $C$ condition number of $H+\sigma^* I$.*

*Proof.* By Theorem 4.1.1, $(H+\sigma^*)x^* + g = 0$. Recall that, by the shift-invariance of Krylov subspaces, $\mathcal{S}_k = \mathcal{K}_k(C^{-1}(H+\sigma^* I), C^{-1}g)$. Let $Q$, $S$, and $\Lambda$ be defined as in Lemma 4.5.10. Then

$$\begin{aligned}
\|(I-\pi_k)x^*\|_C &= \min_{x\in\mathcal{K}_k(C^{-1}(H+\sigma^*), C^{-1}g)} ||x^*-x||_C \\
&= \min_{q\in P_{k-1}} ||x^* - q\big(C^{-1}(H+\sigma^* I)\big)C^{-1}g||_C \\
&= \min_{q\in P_{k-1}} ||x^* + q\big(C^{-1}(H+\sigma^* I)\big)C^{-1}(H+\sigma^* I)x^*||_C \\
&= \min_{p_k\in P_k, p_k(0)=1} ||p_k\big(C^{-1}(H+\sigma^* I)\big)x^*||_C \\
&\le ||x^*||_C \min_{p_k\in P_k, p_k(0)=1} ||p_k\big(\frac{\sigma^*}{\mu}I + \frac{\mu-\sigma^*}{\mu}\Lambda\big)|| \\
&= ||x^*||_C\, \varepsilon_1^{(k)},
\end{aligned}$$

where the last equality is derived via the standard estimates applied to $\varepsilon_1^{(k)}$ using scaled and shifted Chebyshev polynomials of the first kind. □

Now, $y_1$ and $x^*$ only differ in scaling. Thus, the above theorem establishes an upper bound for

$\|(I - \pi_k)y_1\|_C$.

**Corollary 4.5.10.1.** *Let $y = (y_1, y_2)$ be the eigenvector of $M$ associated with $\mu_1 = \sigma^*$. Then*

$$\|(I - \pi_k)y_1\|_C \leq 2||y_1||_C \left( \frac{\sqrt{\kappa_C(H + \sigma^* I)} - 1}{\sqrt{\kappa_C(H + \sigma^* I)} + 1} \right)^{k+1}.$$

So far, the proof has been mostly identical to standard proofs of the convergence of the preconditioned conjugate-gradient algorithm. Unfortunately, in order to analyze the convergence of $\sigma_k$ to $\sigma^*$, the quantity $||(I - \pi_k)y_2||_C$ must be analyzed as well. First, to simplify the notation, let

$$\kappa = \kappa_C(H + \sigma^* I).$$

**Theorem 4.5.11.** *It holds that*

$$||(I - \pi_k)y_2||_C \leq 4 \frac{\tilde{\lambda}_1}{(\tilde{\lambda}_1 - \tilde{\lambda}_n)^2} ||y_1||_C \varepsilon_2^{(k)},$$

*where*

$$\varepsilon_2^{(k)} = \min_{q \in P_{k-1}} \max_{x \in [-1,1]} \left| \frac{1}{(x - \eta)^2} - q(x) \right|,$$

*and*

$$\eta = \frac{\kappa + 1}{\kappa - 1} > 1.$$

*Proof.* Let $Q$, $S$, and $\tilde{\Lambda}$ be defined as in Lemma 4.5.10. From Theorem 4.1.1, $(H + \sigma^* I)x^* + g = 0$. Combining this with Theorem 4.5.7 yields

$$\frac{\delta^2}{g^{\mathrm{T}} y_1}(H + \sigma^*)y_1 = g.$$

The structure of the matrix $N$ yields

$$y_2 = (H + \sigma^* I)^{-1} y_1.$$

Making use of the shift-invariance of Krylov subspaces and the $C$-orthogonality of the matrix $Q$ in the

generalized spectral decomposition of $H + \sigma^* I$, it holds that

$$
\begin{aligned}
||(I - \pi_k)y_2||_C &= \min_{z \in \mathcal{K}_k(C^{-1}(H+\sigma^* I), C^{-1}g)} ||y_2 - z||_C \\
&= \min_{q \in P_{k-1}} ||y_2 - q\big(C^{-1}(H + \sigma^* I)\big)C^{-1}g||_C \\
&= \min_{q \in P_{k-1}} ||(H + \sigma^* I)^{-1}y_1 - \frac{\delta^2}{g^{\mathrm{T}}y_1} q\big(C^{-1}(H + \sigma^* I)\big)C^{-1}(H + \sigma^* I)y_1||_C \\
&= \min_{q \in P_{k-1}} ||(H + \sigma^* I)^{-1}y_1 - \frac{\delta^2}{g^{\mathrm{T}}y_1} C^{-1}(H + \sigma^* I)q\big(C^{-1}(H + \sigma^* I)\big)y_1||_C \\
&= \min_{p \in P_{k-1}} ||C^{-1}(H + \sigma^* I)\left[(H + \sigma^* I)^{-1}C(H + \sigma^* I)^{-1} - p\big(C^{-1}(H + \sigma^* I)\big)\right] y_1||_C \\
&\leq ||H + \sigma^* I||_C \min_{p \in P_{k-1}} ||\left[(H + \sigma^* I)^{-1}C(H + \sigma^* I)^{-1} - p\big(C^{-1}(H + \sigma^* I)\big)\right] y_1||_C \\
&= ||H + \sigma^* I||_C \min_{p \in P_{k-1}} ||Q\left[\tilde{\Lambda}^{-2} - p\big(\tilde{\Lambda}\big)\right] Q^{\mathrm{T}} y_1||_C \\
&\leq \tilde{\lambda}_1 ||y_1||_C \min_{p \in P_{k-1}} \max_{z \in [\tilde{\lambda}_n, \tilde{\lambda}_1]} \left| \frac{1}{z^2} - p(z) \right|.
\end{aligned}
$$

Consider the change of variables

$$z = \frac{\tilde{\lambda}_1 - \tilde{\lambda}_n}{2} x + \frac{\tilde{\lambda}_n + \tilde{\lambda}_1}{2},$$

which creates a bijection from $[-1, 1]$ to $[\tilde{\lambda}_n, \tilde{\lambda}_1]$. Then

$$
\begin{aligned}
&\min_{p \in P_{k-1}} \max_{z \in [\tilde{\lambda}_n, \tilde{\lambda}_1]} \left| \frac{1}{z^2} - p(z) \right| \\
&\quad = \min_{p \in P_{k-1}} \max_{x \in [-1,1]} \left| \frac{4}{(\tilde{\lambda}_1 - \tilde{\lambda}_n)^2 (x - \eta)^2} - p(x) \right| \\
&\quad = \frac{4}{(\tilde{\lambda}_1 - \tilde{\lambda}_n)^2} \min_{p \in P_{k-1}} \max_{x \in [-1,1]} \left| \frac{1}{(x - \eta)^2} - \frac{(\tilde{\lambda}_1 - \tilde{\lambda}_n)^2}{4} p(x) \right| \\
&\quad = \frac{4}{(\tilde{\lambda}_1 - \tilde{\lambda}_n)^2} \min_{q \in P_{k-1}} \max_{x \in [-1,1]} \left| \frac{1}{(x - \eta)^2} - q(x) \right| \\
&\quad = \frac{4}{(\tilde{\lambda}_1 - \tilde{\lambda}_n)^2} \varepsilon_2^{(k)}.
\end{aligned}
$$

□

Unlike $\varepsilon_1^{(k)}$, $\varepsilon_2^{(k)}$ does not have a well-known explicit solution. However, if it can be shown that it is of the same order $\varepsilon_1^{(k)}$, then the proof can proceed. Following [23], Chebyshev polynomials of the second kind shall be used to establish a similar bound.

**Theorem 4.5.12.** *The approximation error satisfies*

$$\varepsilon_2^{(k)} \leq \left(1 + \frac{k+2}{|\ln t|}\right) \frac{4}{1-t^2} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{k+3}$$

*and* $||(I-\pi_k)y_2||_C$ *satisfies*

$$||(I-\pi_k)y_2||_C \leq \frac{16\tilde{\lambda}_1||y_1||_C}{(\tilde{\lambda_1}-\tilde{\lambda_2})^2(1-t^2)} \left(1 + \frac{k+2}{|\ln t|}\right) \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{k+3},$$

*where* $t = \eta - \sqrt{\eta^2-1}$.

*Proof.* Let $U_j(x) = \sin(j \arccos(x))$ denote the $j$-th degree Chebyshev polynomial of the second kind. Then $U_j$ has the following generating function, see [3]:

$$\sum_{j=0}^{\infty}(j+1)t^j U_j(x) = \frac{1-t^2}{(1+t^2-2tx)^2}. \tag{4.34}$$

If $t = \eta - \sqrt{\eta^2-1}$, then it is easily confirmed that $1+t^2 = 2\eta t$. Then (4.34) becomes

$$\sum_{j=0}^{\infty}(j+1)t^j U_j(x) = \frac{1-t^2}{4t^2(x-\eta)^2},$$

or equivalently

$$\frac{1}{(x-\eta)^2} = \frac{4t^2}{1-t^2}\sum_{j=0}^{\infty}(j+1)t^j U_j(x).$$

Let

$$p_k(x) = \frac{4t^2}{1-t^2}\sum_{j=0}^{k}(j+1)t^j U_j(x).$$

Note that $p_k$ is in fact a $k$-th degree polynomial, despite the $1/(1-t^2)$ term. Furthermore, note that

$-\ln t = |\ln t|$ for $t \in (0,1)$ and $|U_j(x)| \leq 1$ for $x \in [-1,1]$. Then

$$\begin{aligned}
\varepsilon_2^{(k)} &\leq \max_{x\in[-1,1]} \left| \frac{1}{(x-\eta)^2} - p_k(x) \right| \\
&= \max_{x\in[-1,1]} \left| \frac{4t^2}{1-t^2} \sum_{j=k+1}^{\infty} (j+1)t^j U_j(x) \right| \\
&\leq \frac{4t^2}{1-t^2} \sum_{j=k+1}^{\infty} (j+1)t^j \\
&= \frac{4t^2}{1-t^2} \int_{k+1}^{\infty} (z+1)t^z dz \\
&= \frac{4t^2}{1-t^2} \left( \frac{z+1}{\ln t} t^z \Big|_{k+1}^{\infty} - t^z \Big|_{k+1}^{\infty} \right) \\
&= \left(1 - \frac{k+2}{\ln t}\right) \frac{4t^{k+3}}{1-t^2} \\
&= \left(1 + \frac{k+2}{|\ln t|}\right) \frac{4t^{k+3}}{1-t^2}.
\end{aligned}$$

As before, $\eta = \frac{\kappa+1}{\kappa-1}$, thus $t = \eta - \sqrt{\eta^2-1} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$. Thus, the result holds. □

Combining the above results yields the following bounds.

**Theorem 4.5.13.** *Suppose that* $||x^*|| = ||x_k|| = \delta$. *Then*

$$\sin\angle(y, \tilde{S}_k)_C \leq c_k ||y_1|| \left( \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^{k+1}$$

*and*

$$\sigma^* - \sigma_k \leq c_k s(\sigma_k) \tilde{\gamma}_k ||y_1|| \left( \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^{k+1},$$

*where*

$$c_k = 2 + \frac{16\tilde{\lambda}_1}{(\tilde{\lambda}_1 - \tilde{\lambda_n})^2(1-t^2)} \left(1 + \frac{k+2}{|\ln t|}\right) \left( \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^2.$$

This establishes that $\sigma_k \to \sigma^*$ as $k$ increases. However, experimentally, it is quite clear that this bound is not particularly sharp. To improve upon this result, the following technical result is derived:

**Theorem 4.5.14.** *For* $k = 0, \ldots, k^*$, *the following bound holds:*

$$e_1^{\mathrm{T}}(D_{k^*} + \sigma^\star T_{k^*})^{-1} e_1 - e_1^{\mathrm{T}}(D_k + \sigma^\star T_k)^{-1} e_1 \leq \frac{4\delta||g||}{\beta^2} \left( \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^{2(k+1)}.$$

*Proof.* Suppose, in the SIGLTR algorithm, $\sigma^*$ and $k^*$ are known *a priori*. Then, at iteration $k^*$, the

projected trust-region subproblem (4.20) could be replaced with the linear system

$$(D_{k^*} + \sigma^\star T_{k^*})h = -\beta e_1, \tag{4.35}$$

where $\beta = ||g||_{C^{-1}}$, and $D_{k^*} + \sigma^\star T_{k^*}$ is symmetric positive definite (by the assumption that the problem in question is not an instance of the hard-case). The linear system implicitly solved at each previous iteration would be

$$(D_k + \sigma^* T_k)\tilde{y}_k = -\beta e_1.$$

Let $E_k = (e_1, e_2, \ldots, e_{k+1})$. Let $h_k = E_k \tilde{y_k}$. Define $\varepsilon_k = h - h_k$, and the residual vector $r_k = -\beta e_1 - (D_{k^*} + \sigma^* T_{k^*})h_k$. Thus,

$$(D_{k^*} + \sigma^* T_{k^*})\varepsilon_k = r_k,$$

and $||r_0|| = \beta^2$. Therefore,

$$\begin{aligned} ||\varepsilon_k||^2_{D_{k^*}+\sigma^* T_{k^*}} &= \varepsilon_k^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})\varepsilon_k \\ &= r_k^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} r_k \\ &= \beta^2(e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 - e_1^{\mathrm{T}}(D_k + \sigma^* T_k)^{-1} e_1), \end{aligned}$$

so

$$e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 - e_1^{\mathrm{T}}(D_k + \sigma^* T_k)^{-1} e_1 = \frac{||\varepsilon_k||^2_{D_{k^*}+\sigma^* T_{k^*}}}{\beta^2}. \tag{4.36}$$

By construction, the eigenvalues of $D_{k^*} + \sigma^* T_{k^*}$ are a subset of the eigenvalues of $(H + \sigma^* I, C)$, thus the eigenvalues of $D_{k^*} + \sigma^* T_{k^*}$ lie in the interval $[\tilde{\lambda}_n, \tilde{\lambda}_1]$. The error $\varepsilon_k$ and residuals $r_k$ are exactly the error and residual vectors of the conjugate-gradient iterations applied to (4.35). Therefore,

$$||\varepsilon_k||^2_{D_{k^*}+\sigma^* T_{k^*}} \leq 4\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2(k+1)} ||\varepsilon_0||^2_{D_{k^*}+\sigma^* T_{k^*}}.$$

The initial error is given by

$$||\varepsilon_0||^2_{D_{k^*}+\sigma^* T_{k^*}} = \beta^2 e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1,$$

Now, $\beta||(D_{k^*} + \sigma^* T_{k^*})e_1||_{T_{k^*}} = ||h||_{T_{k^*}} = \delta$, so

$$\beta^2 e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 \leq \beta||e_1||_{T_{k^*}^{-1}}\delta.$$

Assume that SIGLTR runs, in exact precision, until breakdown occurs. Thus, at iteration $k^*$, it holds that

$$U_{k^*} = V_{k\star} T_{k^*}, \quad \text{or} \quad U_{k^*} T_{k^*}^{-1} = V_{k\star},$$

so that $T_{k^*}^{-1} = V_{k^*}^{\mathrm{T}} V_{k^*} = U_{k^*}^{\mathrm{T}} C^2 U_{k^*}$. Therefore $e_1^{\mathrm{T}} T_{k^*}^{-1} e_1 = g^{\mathrm{T}} g / \beta^2$, and

$$\beta^2 e_1^{\mathrm{T}} (D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 \leq ||g|| \delta.$$

The result then follows. □

The left-hand side of the inequality in Theorem 4.5.14 can be utilized in an upper bound for $\sigma^* - \sigma_k$.

**Theorem 4.5.15.** *Assume that the shifted and inverted Lanczos process breaks down at iteration $k^*$ and that $||x^*|| = ||x_k|| = \delta$ for $k = 0, \ldots, k^*$. Then for $k$ sufficiently large,*

$$\sigma^* - \sigma_k \leq \eta_{k,1} \big( e_1^{\mathrm{T}} (D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 - e_1^{\mathrm{T}} (D_k + \sigma^* T_k)^{-1} e_1 \big) + \eta_{k,2} \big( q(x_k) - q(x^*) \big),$$

*where the scalars $\eta_{k,1}$ and $\eta_{k,2}$ are given by*

$$\eta_{k,1} = \frac{\beta^2}{\delta^2 + \beta^2 e_1^{\mathrm{T}} (D_k + \sigma^* T_k)^{-1} T_k (D_k + \sigma^* T_k)^{-1} e_1},$$

*and*

$$\eta_{k,2} = \frac{2}{\delta^2 + \beta^2 e_1^{\mathrm{T}} (D_k + \sigma^* T_k)^{-1} T_k (D_k + \sigma^* T_k)^{-1} e_1},$$

*and $\beta = ||g||_{(H+\mu I)^{-1}}$.*

*Proof.* At iteration $k$, the solution to the projected trust-region subproblem is

$$h_k = -\beta (D_k + \sigma_k T_k)^{-1} e_1,$$

and $||h_k||_{T_k} = \beta||(D_k + \sigma_k T_k)^{-1}e_1||_{T_k} = \delta$. By (4.20), $q(x_k) = \phi(h_k)$, and

$$\begin{aligned}
q(x_k) &= -\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}e_1 + \frac{1}{2}\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}D_k(D_k + \sigma_k T_k)^{-1}e_1 \\
&= -\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}e_1 \\
&\quad + \frac{1}{2}\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}(D_k + \sigma_k T_k - \sigma_k T_k)(D_k + \sigma_k T_k)^{-1}e_1 \\
&= -\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}e_1 + \frac{1}{2}\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}e_1 \\
&\quad - \frac{1}{2}\sigma_k\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}T_k(D_k + \sigma_k T_k)^{-1}e_1 \\
&= -\frac{1}{2}\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}e_1 - \frac{1}{2}\sigma_k\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}T_k(D_k + \sigma_k T_k)^{-1}e_1 \\
&= -\frac{1}{2}\beta^2 e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}e_1 - \frac{1}{2}\sigma_k\delta^2.
\end{aligned} \tag{4.37}$$

Now, by assumption,

$$x_{k^*} = U_{k^*}h_{k^*} = x^*, \quad \sigma_{k^*} = \sigma^*, \quad \text{and} \quad q(x_{k^*}) = q(x^*) = \phi(h_{k^*}),$$

with $||h_{k^*}||_{T_k^*} = \delta$, and the eigenvalues of $D_{k^*}$ are a subset of the eigenvalues of $(H, C)$. By letting $k = k^*$ in (4.37),

$$q(x^*) = -\frac{1}{2}\beta^2 e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1}e_1 - \frac{1}{2}\sigma^*\delta^2. \tag{4.38}$$

Subtracting (4.37) and (4.38) yields

$$(\sigma^* - \sigma_k)\delta^2 = \beta^2\big(e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}e_1 - e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1}e_1\big) + 2\big(q(x_k) - q(x^*)\big).$$

Consider the expression $e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1}e_1$. It holds that

$$\begin{aligned}
(D_k + \sigma_k T_k)^{-1} &= (D_k + \sigma^* T_k + (\sigma_k - \sigma^*)T_k)^{-1} \\
&= \big((D_k + \sigma^* T_k)(I - (\sigma^* - \sigma_k)(D_k + \sigma^* T_k)^{-1}T_k)\big)^{-1} \\
&= (I - (\sigma^* - \sigma_k)(D_k + \sigma^* T_k)^{-1}T_k)^{-1}(D_k + \sigma^* T_k)^{-1}.
\end{aligned}$$

Consider the matrix $(I - (\sigma^* - \sigma_k)(D_k + \sigma^* T_k)^{-1}T_k)^{-1}$. If $||(\sigma^* - \sigma_k)(D_k + \sigma^* T_k)^{-1}T_k|| < 1$, a Neumann series can be used to analyze $(I - (\sigma^* - \sigma_k)(D_k + \sigma^* T_k)^{-1}T_k)^{-1}$. It has been shown that $(\sigma^* - \sigma_k)$ is nonnegative and tends towards zero as $k \to 0$, so it suffices to show that $||(D_k + \sigma^* T_k)^{-1}T_k||$ is bounded above by some constant. Note that $D_k + \sigma^* T_k = I + (\sigma^* - \mu)T_k$ and $T_k$ commute, therefore $(D_k + \sigma^* T_k)^{-1}$ and $T_k$ commute. $T_k$ is positive definite, thus $T_k$ has a unique positive definite square root $T_k^{1/2}$, which

also commutes with $(D_k + \sigma^* T_k)^{-1}$. So,

$$\begin{aligned}
||(D_k + \sigma^* T_k)^{-1} T_k|| &= ||T_k^{1/2}(D_k + \sigma^* T_k)^{-1} T_k^{1/2}|| \\
&= \max_{x \neq 0} \frac{x^{\mathrm{T}} T_k^{1/2}(D_k + \sigma^* T_k)^{-1} T_k^{1/2} x}{x^{\mathrm{T}} x} \\
&= \max_{y \neq 0} \frac{y^{\mathrm{T}}(D_k + \sigma^* T_k)^{-1} y^{\mathrm{T}}}{y T_k^{-1} y} \\
&\leq \frac{1}{\alpha_n + \sigma^*},
\end{aligned}$$

where $\alpha_n$ is the smallest standard eigenvalue of $H$. Therefore, in order to apply a Neumann series, $k$ needs to be sufficiently large so that $(\sigma^* - \sigma_k)||(D_k + \sigma^* T_k)^{-1} T_k|| < 1$, i.e., $\sigma^* - \sigma_k < \alpha_n + \sigma^*$. By Theorem 4.5.13, a sufficient choice of $k$ is an index such that

$$c_k s(\sigma_k) \tilde{\gamma_k} ||y_1|| \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{k+1} \leq \alpha_n + \sigma^*.$$

By continuity, as $\sigma_k \to \sigma^*$,

$$e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1} e_1 - e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 \to e_1^{\mathrm{T}}(D_k + \sigma^* T_k)^{-1} e_1 - e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1.$$

By (4.36), $e_1^{\mathrm{T}}(D + \sigma_k T_k)^{-1} e_1 - e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1$ must become nonpositive for $k$ sufficiently large. Thus,

$$(\sigma^* - \sigma_k)\delta^2 \leq \beta^2 \big( e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 - e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1} e_1 \big) + 2\big( q(x_k) - q(x^*) \big) \tag{4.39}$$

when $k$ is sufficiently large.

Now, consider the term $e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1} e_1$. As $(\sigma^* - \sigma_k)||(D_k + \sigma_k T_k)^{-1} T_k|| < 1$, the Neumann series of the matrix $(I - (\sigma^* - \sigma_k)(D_k + \sigma_k T_k)^{-1} T_k)^{-1}$ can be considered.

$$\begin{aligned}
(D_k + \sigma_k T_k)^{-1} &= (I - (\sigma^* - \sigma_k)(D_k + \sigma^* T_k)^{-1} T_k)^{-1} (D_k + \sigma^* T_k)^{-1} \\
&= (I + (\sigma^* - \sigma_k)(D_k + \sigma^* T_k)^{-1} T_k + \mathcal{O}((\sigma^* - \sigma_k)^2))(D_k + \sigma^* T_k)^{-1} \\
&= (D_k + \sigma^* T_k)^{-1} + (\sigma^* - \sigma_k)(D_k + \sigma^* T_k)^{-1} T_k (D_k + \sigma^* T_k)^{-1} \\
&\quad + \mathcal{O}((\sigma^* - \sigma_k)^2).
\end{aligned}$$

Therefore,

$$\begin{aligned} & e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 - e_1^{\mathrm{T}}(D_k + \sigma_k T_k)^{-1} e_1 \\ & \quad = e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 - e_1^{\mathrm{T}}(D_k + \sigma^* T_k)^{-1} e_1 \\ & \quad \quad - (\sigma^* - \sigma_k) e_1^{\mathrm{T}}\big((D_k + \sigma^* T_k)^{-1} T_k (D_k + \sigma^* T_k)^{-1}\big) e_1 \\ & \quad \quad - \mathcal{O}\big((\sigma^* - \sigma_k)^2\big), \end{aligned} \tag{4.40}$$

which is nonnegative if $k$ is sufficiently large. Substituting this into (4.39) and dropping higher order terms gives

$$\sigma^* - \sigma_k \leq \eta_{k,1}\big(e_1^{\mathrm{T}}(D_{k^*} + \sigma^* T_{k^*})^{-1} e_1 - e_1^{\mathrm{T}}(D_k + \sigma^* T_k)^{-1} e_1\big) + \eta_{k,2}\big(q(x_k) - q(x^*)\big),$$

proving the result. Note that both $\eta_{k,1}$ and $\eta_{k,2}$ can be bounded above for $k$ sufficiently large. □

All that remains now is to bound $q(x_k) - q(x^*)$.

**Theorem 4.5.16.** *Suppose that $||x^*|| = ||x_k|| = \delta$. Let $c_1, c_2 > 0$ be constants such that*

$$c_1||x|| \leq ||x||_C \leq c_2||x||,$$

*and*

$$\frac{1}{c_2}||x||_C \leq ||x|| \leq \frac{1}{c_1}||x||_C$$

*for all $x$. Then,*

$$0 \leq q(x_k) - q(x^*) \leq \frac{(1 + \frac{c_2}{c_1})^2}{2} \tilde{\lambda}_1 ||\tilde{x} - x^*||_C^2 \tag{4.41}$$

*for any $\tilde{x} \in \mathcal{K}_k(C^{-1}H, C^{-1}g)$ not equal to zero.*

*Proof.* Let $\tilde{x} \in \mathcal{K}_k(C^{-1}H, C^{-1}g)$ and $\tilde{x} \neq 0$. Let $v = \frac{\delta}{||\tilde{x}||}\tilde{x}$, and $r = v - x^*$. Then,

$$\begin{aligned} 0 \leq q(x_k) - q(x^*) & \leq q(v) - q(x^*) \\ & = \frac{1}{2} r^{\mathrm{T}} H r + r^{\mathrm{T}}(Hx^* + g) \\ & = \frac{1}{2} r^{\mathrm{T}} H r - \sigma^* r^{\mathrm{T}} x^* \\ & = \frac{1}{2} r^{\mathrm{T}}(H + \sigma^* I) r \leq \frac{1}{2} ||H + \sigma^* I||_C ||r||_C^2. \end{aligned}$$

Now,

$$\begin{aligned}
||r||_C &\leq ||x^* - \tilde{x}||_C + ||\tilde{x} - v||_C \\
&= ||x^* - \tilde{x}||_C + ||\tilde{x} - \frac{\delta}{||\tilde{x}||_2}\tilde{x}||_C \\
&= ||x^* - \tilde{x}||_C + ||\tilde{x}||_C \left|1 - \frac{\delta}{||\tilde{x}||_2}\right| \\
&\leq ||x^* - \tilde{x}||_C + c_2||\tilde{x}|| \left|1 - \frac{\delta}{||\tilde{x}||_2}\right| \\
&\leq ||x^* - \tilde{x}||_C + c_2\big|||\tilde{x}|| - \delta\big| \\
&= ||x^* - \tilde{x}||_C + c_2\big|||\tilde{x}|| - ||x^*|||\big| \\
&\leq ||x^* - \tilde{x}||_C + c_2||\tilde{x} - x^*|| \\
&\leq ||x^* - \tilde{x}||_C + \frac{c_2}{c_1}||\tilde{x} - x^*||_C \\
&= (1 + \frac{c_2}{c_1})||x^* - \tilde{x}||_C.
\end{aligned}$$

Thus,

$$q(x_k) - q(x^*) \leq \frac{(1 + \frac{c_2}{c_1})^2}{2}||H + \sigma^* I||_C||\tilde{x} - x^*||_C^2.$$

□

Using bounds for $||x_k - x^*||_C$ yields the following result.

**Theorem 4.5.17.** *Suppose* $||x^*|| = ||x_k|| = \delta$. *Then*

$$0 \leq q(x_k) - q(x^*) \leq 2(1 + \frac{c_2}{c_1})^2 c_2^2 \delta^2 \tilde{\lambda}_1 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^{2(k+1)}.$$

*Proof.* Equation (4.41) gives

$$0 \leq q(x_k) - q(x^*) \leq \frac{(1 + \frac{c_2}{c_1})^2}{2}\tilde{\lambda}_1 \min_{\tilde{x} \in \mathcal{K}_k(C^{-1}H, C^{-1}g)} ||\tilde{x} - x^*||_C^2.$$

Recall that

$$\min_{\tilde{x} \in \mathcal{K}_k(C^{-1}H, C^{-1}g)} ||\tilde{x} - x^*||_C^2 = ||(I - \pi_k)x^*||_C^2.$$

Combining this with lemma (4.5.10) gives

$$\min_{\tilde{x} \in \mathcal{K}_k(C^{-1}H, C^{-1}g)} ||\tilde{x} - x^*||_C^2 \leq 4||x^* \leq 4||x^*||_C^2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^{2(k+1)}.$$

Using $||x^*||_C^2 \leq c_2^2\delta^2$ yields the result. □

The quantity $\sigma^* - \sigma^k$ can now be bounded.

**Theorem 4.5.18.** *Suppose that* $||x^*|| = ||x_k|| = \delta$. *Then, if* $k$ *is sufficiently large,*

$$\sigma^* - \sigma_k \leq \left( \frac{4\eta_{k,1}||g||\delta}{\beta^2} + 2\eta_{k,2}(1 + \frac{c_2}{c_1})^2 c_2^2 \delta^2 \tilde{\lambda}_1 \right) \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2(k+1)}.$$

This theorem indicates that $\sigma^* - \sigma_k$ converges at least as fast as $\left( \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^{2(k+1)}$. Thus, as expected, $\sigma_k$ converges to $\sigma^*$ significantly faster than $x_k$ converges to $x^*$. The multiplier $\sigma_k$ will converge to some level of accuracy in about half as many iterations as $x_k$ will need to reach the same level of accuracy. This indicates that a restarting technique could be used in which SIGLTR is restarted with a new shift $\mu_2 = \sigma_k$.

### 4.5.6 Effect of Shifting on the Convergence Rate

This section analyzes the effect of the choice of $\mu$ on the convergence rate. Consider the generalized eigenvalue problem

$$Hx = \lambda Mx,$$

with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Then the problem

$$(H + \sigma M)x = \lambda Mx,$$

has eigenvalues $\lambda_1 + \sigma \geq \lambda_2 + \sigma \geq \ldots \geq \lambda_n + \sigma$. Now, consider

$$(H + \sigma M)x = \nu(H + \mu M)x.$$

The eigenvalues $\nu_i$ of this problem can be expressed in terms of $\lambda_1, \ldots, \lambda_n$. So,

$$Hx = \frac{\nu\mu - \sigma}{1 - \nu} Mx.$$

Thus, if $\nu$ is an eigenvalue of the above problem, $\lambda = (\nu\mu - \sigma)/(1 - \nu)$ is an eigenvalue of the original problem. A simple rearrangement of this expression gives

$$\nu = \frac{\lambda + \sigma}{\lambda + \mu}.$$

Note that by assumption, $\lambda_j + \mu > 0$ for all $j \in \{1, \ldots, n\}$. Thus, if $\sigma \leq \mu$, the $(H + \mu M)$-condition number is given by

$$\kappa_{H+\mu M}(H + \sigma M) = \frac{\nu_1}{\nu_n} = \frac{(\lambda_1 + \sigma)/(\lambda_1 + \mu)}{(\lambda_n + \sigma)/(\lambda_n + \mu)} = \frac{\kappa_M(H + \sigma M)}{\kappa_M(H + \mu M)}.$$

Conversely, if $\mu < \sigma$, then

$$\kappa_{H+\mu M}(H + \sigma M) = \frac{\nu_1}{\nu_n} = \frac{(\lambda_n + \sigma)/(\lambda_n + \mu)}{(\lambda_1 + \sigma)/(\lambda_1 + \mu)} = \frac{\kappa_M(H + \mu M)}{\kappa_M(H + \sigma M)}.$$

Thus,

$$\kappa_{H+\mu M}(H + \sigma M) = \frac{\kappa_M(H + \min\{\sigma, \mu\}M)}{\kappa_M(H + \max\{\sigma, \mu\}M)}. \tag{4.42}$$

Consider the the case where $\mu$ is very close to $\sigma$. More specifically, consider the case $0 < \mu - \sigma \leq \varepsilon(\lambda_1 - \lambda_n)$. Then

$$\begin{aligned}
\kappa_{H+\mu M}(H + \sigma M) &= \kappa_M(H + \sigma M)\frac{\lambda_n + \mu}{\lambda_1 + \mu} \\
&= \kappa_M(H + \sigma M)\frac{\lambda_n + \sigma + \mu - \sigma}{\lambda_1 + \sigma + \mu - \sigma} \\
&\leq \kappa_M(H + \sigma M)\frac{\lambda_n + \sigma + \varepsilon(\lambda_1 - \lambda_n)}{\lambda_1 + \sigma} \\
&\leq \kappa_M(H + \sigma M)(\frac{1}{\kappa_M(H + \sigma M)} + \varepsilon\frac{\lambda_1 - \lambda_n}{\lambda_1 + \sigma}).
\end{aligned}$$

By assumption, $-\lambda_n \leq \sigma$, so it holds that

$$1 \leq \kappa_{H+\mu M}(H + \sigma M) \leq 1 + \varepsilon\kappa_M(H + \sigma M).$$

Similarly, if $0 < \sigma - \mu < \varepsilon(\lambda_1 - \lambda_n)$, then

$$1 \leq \kappa_{H+\mu M}(H + \sigma M) \leq 1 + \varepsilon\kappa_M(H + \mu M).$$

Thus, if $|\mu - \sigma| \leq \varepsilon(\lambda_1 - \lambda_n)$

$$1 \leq \kappa_{H+\mu M}(H + \sigma M) \leq 1 + \varepsilon\kappa_M(H + \min\{\sigma, \mu\}M).$$

Thus, it can be seen here that it is essential to choose a $\mu$ that is not too close to $\lambda_n$. Otherwise, the matrix $H + \mu M$ becomes nearly singular. If the trust-region radius is relatively small, then this will lead

to ill-conditioning.

### 4.5.7 Warm-starting and Restarting

Consider using PCG, or equivalently Lanczos-CG, to solve the problem $Hx = b$, where some vector $x_0$ is known such that the residual vector $r = b - Ax_0$ has $||r|| \ll ||b||$, i.e., $x_0$ is an initial approximation to the solution $x$ of $Hx = b$. Then the linear system can be reformulated as $H(x_0 + \bar{x}) = b$, or equivalently $H\bar{x} = r$, where the solution to the original problem is then $x = x_0 + \bar{x}$, and $\bar{x}$ can be solved for via PCG. Such a procedure can be added to an implementation of Algorithm 1.1 to either warm-start PCG or restart PCG when the search directions lose their $H$-conjugacy (or equivalently, when the residual vectors lose their $M$-orthogonality). A similar technique can be applied to SIGLTR. It is unknown if any similar procedure has been used with GLTR. However, what follows would only require minor modifications to apply to GLTR.

Consider an initial approximation $x_0$ to the solution of the trust-region problem (4.1). No restrictions are placed on $x_0$, though for practical reasons, $x_0$ should be scaled to lie within the trust region. Problem (4.1) can be written as

$$\begin{aligned} \min_{\bar{x} \in \mathbb{R}^n} \ & \frac{1}{2}(\bar{x} + x_0)^{\mathrm{T}} H \bar{x} + g^{\mathrm{T}}(\bar{x} + x_0) \\ \text{subject to} \quad & ||\bar{x} + x_0||_M \leq \delta, \end{aligned} \tag{4.43}$$

or, after removing constant terms from the objective, as

$$\begin{aligned} \min_{\bar{x} \in \mathbb{R}^n} \ & \frac{1}{2}\bar{x}^{\mathrm{T}} H \bar{x} + (g + Hx_0)^{\mathrm{T}} \bar{x} \\ \text{subject to} \quad & \bar{x}^{\mathrm{T}} M \bar{x} + 2x_0^{\mathrm{T}} M \bar{x} + x_0^{\mathrm{T}} M x_0 \leq \delta^2, \end{aligned} \tag{4.44}$$

Problem (4.44) is not a trust-region problem. Instead, it is an instance of what is referred to as a *generalized trust-region problem*. However, as it is equivalent to a trust-region problem, a simple modification to SIGLTR allows problem (4.44) to be solved without resorting to more complicated methods for solving generalized trust-region problems.

The goal is to apply the block version of SIGLTR to problem (4.44), with a block size of 2. Suppose the initial block is chosen so that $\bar{V}_1 = (g + Hx_0, Mx_0)$. Applying Gram-Schmidt biorthogonalization then yields $U_1$ and $V_1$, where $V_1 B_1 = \bar{V}_1$, $(H + \mu M)U_1 = V_1$, and $B_1 = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} \\ 0 & \beta_{2,2} \end{pmatrix}$. If $U$ is the full

$n \times n$ matrix of Lanczos vectors, then

$$U^{\mathrm{T}}(g + Hx_0) = \begin{pmatrix} \beta_{1,1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \text{and} \quad U^{\mathrm{T}}Mx_0 = \begin{pmatrix} \beta_{1,2} \\ \beta_{2,2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Also note that the initial residual $r_0 = g + Hx_0 + \sigma_0 Mx_0 \in \text{range}(V_1)$ for any value of $\sigma_0$. Projecting the problem onto the $k$-th Krylov subspace then gives

$$\begin{aligned} &\min_{y \in \mathbb{R}^{(2k)}} \ \frac{1}{2} y^{\mathrm{T}}(I - \mu T_k)y + \beta_{1,1} e_1^{\mathrm{T}} y \\ &\text{subject to} \quad y^{\mathrm{T}} T_k y + 2(\beta_{1,2}e_1 + \beta_{2,2}e_2)^{\mathrm{T}} y + (x_0^{\mathrm{T}} M x_0) \leq \delta^2, \end{aligned} \tag{4.45}$$

where the $k$-th approximation to the true solution $x$ is given by $x_k = x_0 + U_k y_k$. The optimality conditions of problem (4.45) are

1. $(I + (\sigma - \mu)T_k)y_k + \beta_{1,1}e_1 + \sigma(\beta_{1,2}e_1 + \beta_{2,2}e_2) = 0$,
2. $\sigma \geq 0$,
3. $y^{\mathrm{T}} T_k y + 2(\beta_{1,2}e_1 + \beta_{2,2}e_2)^{\mathrm{T}} y + (x_0^{\mathrm{T}} M x_0) \leq \delta^2$,
4. $\sigma(\delta^2 - y^{\mathrm{T}} T_k y - 2(\beta_{1,2}e_1 + \beta_{2,2}e_2)^{\mathrm{T}} y - (x_0^{\mathrm{T}} M x_0)) = 0$, and
5. $(I + (\sigma - \mu)T_k) \succeq 0$.

A simple calculation shows that the $(H + \mu M)^{-1}$ norm of the residual vector $r_k = g + Hx_k + \sigma_k Mx_k$ is given by equation (4.29).

The projected subproblem (4.45) is not a trust-region subproblem. It can, however, be manipulated into a trust-region subproblem. Let $b_1 = \beta_{1,1}e_1$ and $b_2 = \beta_{1,2}e_1 + \beta_{2,2}e_2$. The projected constraint can be written as

$$y^{\mathrm{T}} T_k y^{\mathrm{T}} + 2b_2^{\mathrm{T}} y + b_2^{\mathrm{T}} T_k^{-1} b_2 \leq \delta^2 + b_2^{\mathrm{T}} T_k^{-1} b_2 - (x_0^{\mathrm{T}} M x_0),$$

or equivalently

$$(y + T_k^{-1} b_2)^{\mathrm{T}} T_k (y + T_k^{-1} b_2) \leq \delta^2 + b_2^{\mathrm{T}} T_k^{-1} b_2 - (x_0^{\mathrm{T}} M x_0),$$

Let $\bar{\delta} = (\delta^2 + b_2^{\mathrm{T}} T_k^{-1} b_2 - (x_0^{\mathrm{T}} M x_0))^{1/2}$. Then the projected problem can be written as

$$\begin{aligned} \min_{\bar{y} \in \mathbb{R}^{(2k)}} \ & \frac{1}{2} \bar{y}^{\mathrm{T}} (I - \mu T_k) \bar{y} + (b_1 - T_k^{-1} b_2 + \mu b_2)^{\mathrm{T}} \bar{y} \\ \text{subject to} \ & ||\bar{y}||_{T_k} \leq \bar{\delta}, \end{aligned} \tag{4.46}$$

As $T_k$ is a block-tridiagonal matrix, problem (4.46) can be solved efficiently with the Moré-Sorensen algorithm. The vector $y_k$ can be recovered with $y_k = \bar{y}_k - T_k^{-1} b_2$. If $x_0$ is chosen to lie within the trust region, i.e., $||x_0|| \leq \delta$, then $\bar{\delta}$ is guaranteed to be positive.

As with SIGLTR, warm-started SIGLTR can be broken up into two phases. While the approximate solution $x_k = x_0 + U_k y_k$ lies strictly within the trust region, SIGLTR is equivalent to Lanczos-CG applied to the warm-started system $H(x_0 + \bar{x}) = r_0 = -g - H x_0$. All computations for computing $x_k$ remain identical, including the calculation of the residual $r_k (H + \mu M)^{-1} r_k = \xi_{k+1,1}^2 + \xi_{k+1,2}^2$. More precisely, at each iteration, $x_k$ is computed to be $x_k = x_0 + U_k y_k = x_0 + P_k z_k = x_{k-1} + p_{k,1} \xi_{k,1} + p_{k,2} \xi_{k,2}$, where the search-directions $p_{k,1}$ and $p_{k,2}$ are computed in the same manner as before. Using two search directions reflects that the block size was chosen to be two. The only change is in the computation of the norm of the approximation $x_k$. Now, it holds that $x_k M x_k = (x_0 + \bar{x}_k)^{\mathrm{T}} M (x_0 + \bar{x}_k) = x_0^{\mathrm{T}} M x_0 + 2 x_0^{\mathrm{T}} M \bar{x}_k + \bar{x}_k^{\mathrm{T}} M \bar{x}_k$, where $\bar{x}_k = U_k y_k = P_k z_k$. The term $x_0^{\mathrm{T}} M x_0$ can be computed upfront, while $\bar{x}_k^{\mathrm{T}} M \bar{x}_k$ can be computed using block versions of equations (4.23), (4.24), (4.25), and (4.26). All that remains is to show how to compute $x_0^{\mathrm{T}} M \bar{x}_k$. It holds that

$$\begin{aligned} x_0^{\mathrm{T}} M \bar{x}_k = \begin{pmatrix} \beta_{1,2} & \beta_{2,2} \end{pmatrix} V_1^{\mathrm{T}} \bar{x}_k &= \begin{pmatrix} \beta_{1,2} & \beta_{2,2} \end{pmatrix} V_1^{\mathrm{T}} P_k z_k \\ &= x_0^{\mathrm{T}} B \bar{x}_{k-1} + \begin{pmatrix} \beta_{1,2} & \beta_{2,2} \end{pmatrix} V_1^{\mathrm{T}} \begin{pmatrix} p_{k,1} & p_{k,2} \end{pmatrix} \begin{pmatrix} \xi_{k,1} \\ \xi_{k,2} \end{pmatrix} \end{aligned}$$

Now, $x_0^{\mathrm{T}} M \bar{x}_0 = 0$,

$$\begin{pmatrix} p_{1,1} & p_{1,2} \end{pmatrix} = \begin{pmatrix} u_{1,1} & u_{1,2} \end{pmatrix} D_1^{-1},$$

and

$$\begin{pmatrix} p_{k,1} & p_{k,2} \end{pmatrix} = \left( \begin{pmatrix} u_{k,1} & u_{k,2} \end{pmatrix} + \mu \begin{pmatrix} p_{k-1,1} & p_{k-1,2} \end{pmatrix} B_k^{\mathrm{T}} \right) D_k^{-1}$$

for all $k > 1$, and therefore

$$V_1^{\mathrm{T}} \begin{pmatrix} p_{1,1} & p_{1,2} \end{pmatrix} = D_1^{-1}, \quad \text{and} \quad V_1^{\mathrm{T}} \begin{pmatrix} p_{k,1} & p_{k,2} \end{pmatrix} = \mu V_1^{\mathrm{T}} \begin{pmatrix} p_{k-1,1} & p_{k-1,2} \end{pmatrix} B_k^{\mathrm{T}} D_k^{-1}.$$

These equations can update $x_0^{\mathrm{T}} M\bar{x}_k$ without computing additional inner products with vectors of length $n$ or any other matrix-vector multiplications. Thus, SIGLTR can be effectively warm-started or restarted by relaxing the trust-region problem to a generalized trust-region problem. Experiments reveal that the convergence to the true solution $x$ is effected similarly to restarting the preconditioned conjugate-gradient method.

### 4.5.8 Use in a Trust-Region Algorithm

In a typical trust-region algorithm, such as Algorithm 2.2, it is often necessary to solve a sequence of trust-region subproblems where the only difference is the decreased radius. Suppose the GLTR algorithm is used in a case where a particular subproblem needs to be solved multiple times, with a reduced radius in each instance. Recall that GLTR requires access to the matrix-vector operations $v \leftarrow Hu$ and $u \leftarrow M^{-1}v$. In the limit as the radius $\delta$ goes to zero, the solution to the subproblem converges in direction to the steepest-descent direction $-M^{-1}g$. Because the operation $u \leftarrow M^{-1}v$ is assumed to be available, the GLTR algorithm will eventually only require one iteration to converge. This is further motivated by the fact that as $\delta \to 0$, $\sigma \to \infty$ and $\kappa_M(H + \sigma M) \to 1$. This implies that as $\delta \to 0$, GLTR will require fewer and fewer iterations to converge. Observe that the tridiagonal matrix $T$ generated by the GLTR algorithm does not depend on the value of $\delta$. Therefore, it can be reused for each value of $\delta$ without needing to be regenerated. An optimist may look at this situation as a positive. If the subproblem needs to be resolved with a different radius, less work is required. On the other hand, a pessimist may argue the extra computation used to compute the solution of the initial trust-region subproblem is wasted, and that it would be preferable that the initial problem be the simpler problem to solve. This is another benefit that SIGLTR affords.

Suppose a trust-region method arrives at a point where a trust-region subproblem needs to be solved for a sequence of radii $\delta_1 > \delta_2 > \ldots > \delta_{j_{\max}}$. Additionally, suppose some oracle provides the optimal value of the dual variable $\sigma_1$ for the initial trust-region problem, and that the problem in question is not an instance of the hard case. Let $\mu$ be the fixed shift used for each application of SIGLTR, and set $\mu = \sigma_1$. Then the initial problem is solved in one iteration, with the solution given by $-(H + \mu M)^{-1}g$, and $\kappa_{H+\mu M}(H + \sigma_1 M) = 1$. It then holds that $\mu = \sigma_1 < \sigma_2 < \ldots < \sigma_{j_{\max}}$. Moreover, by (4.42), it holds that

$$\kappa_{H+\mu M}(H + \sigma_j M) = \kappa_M(H + \mu M)\frac{\lambda_n + \sigma_j}{\lambda_1 + \sigma_j},$$

where $\lambda_1 > \lambda_n$ are the largest and smallest eigenvalues of $(H, M)$, respectively. This expression is strictly increasing on the domain $\sigma > -\lambda_n$ and is bounded above by $\kappa_M(H + \mu M)$, and therefore

$\kappa_{H+\mu M}(H + \sigma_j M) < \kappa_{H+\mu M}(H + \sigma_{j+1} M) < \kappa_M(H + \mu M)$ for all $j$. This implies that, in general, the number of iterations required by SIGLTR will increase as the radius converges to zero. Of course, no oracle to provide $\mu = \sigma_1$ exists, however, good choices of $\mu$ will exhibit similar asymptotic behavior.

As with GLTR, the matrix $T$ computed by the Lanczos process can be recycled. If the $i$-th vectors used in the Lanczos process are stored, then SIGLTR can begin solving the reduced problem at iteration $i$, and proceed until termination. Thus, no unnecessary Lanczos vectors are computed. A similar argument applies to the warm-started SIGLTR algorithm. Of course, the instability of the Lanczos process still presents a problem, and thus the best results are found by combining the restart strategies of this section and the previous section. To be more precise, for a given trust-region subproblem solved for a decreasing sequence of trust-region radii, an upper limit on the number of SIGLTR iterations should be set. Once this number is reached, the algorithm should be restarted with the technique of the previous section independent of how many times the trust-region subproblem has been solved thus far. This upper limit should depend on whether the Lanczos vectors are stored in memory, whether a reorthogonalization scheme is used, and the estimated conditioning of the problem.

## 4.6 A Jacobi-Davidson Correction Trust-Region Algorithm

One advantage of the shifted and inverted GLTR algorithm is that the subproblem on the basis spanned by the computed Lanczos vectors is increasing in dimension at each iteration. Unlike methods that work with successive subspaces, no information is discarded as the algorithm proceeds. Provided that the choice in shift $\mu$ is reasonably accurate, significantly fewer iterations are required than in standard GLTR. However, there is still the unfortunate need to compute solutions to $(H + \mu M)u = v$ at each iteration. In many large, well-structured problems, sophisticated sparse factorization techniques can mitigate this issue. However, an upper limit of matrix size and density exists where using explicit matrix factorizations becomes infeasible. One reasonable option would be to solve $(H + \mu M)u = v$ at each iteration via an iterative method. As $H + \mu M$ is constructed to be positive definite, the conjugate-gradient or preconditioned conjugate-gradient method is a natural choice. Unfortunately, a tight convergence criterion must be employed to ensure sufficient accuracy. Otherwise, the computed solution will have some error component away from the actual Lanczos vector. This will lead to each iteration requiring many matrix-vector products to calculate the necessary Lanczos vector to high precision.

Furthermore, unlike using an approach based on explicit matrix factors, no work can be preserved between each iteration. One benefit of this approach is that an arbitrary preconditioner of $H + \mu M$ may be used, giving this approach somewhat more flexibility. However, the high degree of accuracy needed

makes this method prohibitively expensive.

Instead, it would be convenient to derive a method that builds up an orthonormal basis of subspaces, like SIGLTR, in which the basis vectors can be computed by solving a linear system with low accuracy. These goals are accomplished by relaxing the requirement that the subspace be a Krylov subspace.

Consider some initial guess of the trust-region problem's primal-dual solution $(x_0, \sigma_0)$, and the corresponding residual vector $r_0 = -g - (H + \sigma_0 M)x_0$. Furthermore, suppose that $x_0$ is scaled such that it is the best approximation in the span of the matrix $V = (v_0)$, and that $\sigma_0$ is the appropriate dual variable, i.e. so that $r_0^{\mathrm{T}} x_0 = 0$. Finally, let $\sigma^*$ be the optimal value of $\sigma$. The motivating idea is to extend the search space by adding a column $v_1$ to the matrix $V$ such that $v_1^{\mathrm{T}} M x_0 = 0$, and that solution in the expanded basis reduces the residual to zero. This ideal vector is constructed by solving the following problem, referred to as the *correction equation*:

$$(H + \sigma^* M)(x_0 + v_1) = -g, \quad v_1^{\mathrm{T}} M x_0 = 0, \tag{4.47}$$

or equivalently

$$(H + \sigma^* M)v_1 = r_0 + (\sigma_0 - \sigma^*)Mx, \quad v_1^{\mathrm{T}} M x_0 = 0.$$

Solving this equation exactly would yield the solution to the trust-region subproblem. However, as $\sigma^*$ is unknown at this point, it is replaced with the best approximation available, i.e., $\sigma_0$, yielding

$$(H + \sigma_0 M)v_1 = r_0, \quad v_1^{\mathrm{T}} M x_0 = 0. \tag{4.48}$$

By the second condition, $v_1 = (I - (x_0 x_0^{\mathrm{T}} M)/(x_0^{\mathrm{T}} M x_0))v_1$, so

$$(H + \sigma_0 M)(I - \frac{x_0 x_0^{\mathrm{T}} M}{x_0^{\mathrm{T}} M x_0})v_1 = r_0.$$

This equation can be projected onto the space spanned by $Mx_0$ and its orthogonal complement.

$$\frac{M x_0 x_0^{\mathrm{T}}}{x_0^{\mathrm{T}} M x_0}(H + \sigma_0 M)(I - \frac{x_0 x_0^{\mathrm{T}} M}{x_0^{\mathrm{T}} M x_0})v_1 = \frac{M x_0 x_0^{\mathrm{T}}}{x_0^{\mathrm{T}} M x_0} r_0 = 0,$$

as $r_0^{\mathrm{T}} x_0 = 0$ by assumption, and

$$(I - \frac{M x_0 x_0^{\mathrm{T}}}{x_0^{\mathrm{T}} M x_0})(H + \sigma_0 M)(I - \frac{x_0 x_0^{\mathrm{T}} M}{x_0^{\mathrm{T}} M x_0})v_1 = (I - \frac{M x_0 x_0^{\mathrm{T}}}{x_0^{\mathrm{T}} M x_0}) r_0 = r_0.$$

Putting everything together, the correction equation is

$$(I - \frac{Mx_0x_0^{\mathrm{T}}}{x_0^{\mathrm{T}}Mx_0})(H + \sigma_0 M)(I - \frac{x_0x_0^{\mathrm{T}}M}{x_0^{\mathrm{T}}Mx_0})v_1 = r_0, \quad v_1^{\mathrm{T}}Mx_0 = 0.$$

This correction equation is similar to the correction equation used in the Jacobi-Davidson method for solving eigenvector problems (see [10]). However, in that case, the residual takes a different form. As it is used here to correct the error in the solution to the trust-region equation, the same terminology is used. It is for this reason that this method is called the Jacobi-Davidson trust-region algorithm (JDTR). Once solved, $v_1$ is added to the search space matrix $V$, and the subproblem

$$\min_{y \in \mathbb{R}^2} \frac{1}{2} y^{\mathrm{T}} V^{\mathrm{T}} H V y + g^{\mathrm{T}} V y$$
$$\text{subject to} \quad ||y||_{V^{\mathrm{T}}MV} \leq \delta,$$

is solved. The solution is approximated as $x_1 = Vy$, and $\sigma_1$ is set to the dual variable of the above problem. The residual is computed to be $r_1 = -g - (H + \sigma_1 M)x_1$. A new correction equation is solved, and $v_2$ is added to $V$. However, as $v_2$ is only guaranteed to be $M$-orthogonal to $x_1$, it is important to use Gram-Schmidt biorthogonalization on $v_2$ against the vectors currently in $V$. Thus, the main computational work at each iteration lies in solving the correction equation

$$(I - \frac{Mx_ix_i^{\mathrm{T}}}{x_i^{\mathrm{T}}Mx_i})(H + \sigma_i M)(I - \frac{x_ix_i^{\mathrm{T}}M}{x_i^{\mathrm{T}}Mx_i})v_{i+1} = r_i, \quad v_{i+1}^{\mathrm{T}}Mx_i = 0, \tag{4.49}$$

and maintaining the $M$-orthogonality of the basis vectors.

A solution $y$ to the subproblem at any iteration $k$ satisfies the equation

$$V^{\mathrm{T}}g + V^{\mathrm{T}}(H + \sigma M)Vy = V^{\mathrm{T}}(g + (H + \sigma M)Vy) = 0.$$

This is equivalent to imposing a Galerkin condition on $x$ at each iteration, i.e.,

$$x \in \text{range}(V), \quad \text{and} \quad g + (H + \sigma M)x \perp V,$$

which enforces the condition $r_k^{\mathrm{T}}x_k = 0$ for all $k$.

A reasonable question would be whether it is crucial to enforce the condition $v_{k+1}^{\mathrm{T}}Mx_k = 0$ when computing search directions. If this condition is dropped, and $H + \sigma M$ is nonsingular (as will most

commonly be the case), then (4.48) becomes

$$v_{i+1} = -(H + \sigma_i M)^{-1} g - x_i.$$

However, if the algorithm is reasonably far along, then $x$ will already be a good approximation of $-(H + \sigma M)^{-1} g$, meaning that very little new information is obtained. If $g = 0$, the solution becomes $v = -x$, and no new information is obtained. Thus, enforcing the condition $v_{k+1}^{\mathrm{T}} M x_k = 0$ ensures that the subspace remains robust and keeps the algorithm from breaking down prematurely. This is particularly important when using floating point arithmetic, as round-off error and the instability of the Gram-Schmidt process will cause the search directions to become linearly dependent. The biggest remaining question is how to solve the correction equation (4.49). If $H + \sigma M$ is positive definite, then $\bar{H} = (I - M x_k x_k^{\mathrm{T}} / (x_k^{\mathrm{T}} M x_k))(H + \sigma M)(I - x_k x_k^{\mathrm{T}} M / (x_k^{\mathrm{T}} M x_k))$ is positive definite on the $M$-orthogonal complement of $x_k$. Therefore, preconditioned conjugate-gradient is a viable choice of algorithm. Let $B \approx H + \sigma_i M$ be some preconditioner for $H + \sigma_i M$. Then

$$\bar{B} = (I - \frac{M x_k x_k^{\mathrm{T}}}{x_k^{\mathrm{T}} M x_k}) B (I - \frac{x_k x_k^{\mathrm{T}} M}{x_k^{\mathrm{T}} M x_k})$$

is a suitable preconditioner for $\bar{H}$. What remains to show is how to solve the equation $\bar{B} u = v$, $u^{\mathrm{T}} M x_k = 0$, where $v^{\mathrm{T}} x_k = 0$. As $u^{\mathrm{T}} M x_k = 0$, $(I - \frac{x_k x_k^{\mathrm{T}} M}{x_k^{\mathrm{T}} M x_k}) u = u$, so

$$(I - \frac{M x_k x_k^{\mathrm{T}}}{x_k^{\mathrm{T}} M x_k}) B u = v.$$

Therefore,

$$u = \alpha B^{-1} M x_k + B^{-1} v.$$

The scalar $\alpha$ can be determined via the condition $u^{\mathrm{T}} M x_k = 0$.

$$0 = x_k^{\mathrm{T}} M u = \alpha x_k^{\mathrm{T}} M B^{-1} M x_k + x_k^{\mathrm{T}} M B^{-1} v,$$

or

$$\alpha = -\frac{x_k^{\mathrm{T}} M B^{-1} v}{x_k^{\mathrm{T}} M B^{-1} M x_k}.$$

Note that $B^{-1} M x_k$ need only be computed once.

Suppose the preconditioned conjugate-gradient algorithm is used to solve $\bar{H}_k v_{k+1} = -r_k$ exactly.

In that case, the Jacobi-Davidson trust-region algorithm will require prohibitively many matrix-vector products per iteration, much like SIGLTR using preconditioned conjugate-gradient to solve for each Lanczos vector. Fortunately, as the Jacobi-Davidson trust-region method does not impose any conditions on the matrix $V$ besides $M$-orthogonality, equation (4.47) need not be solved exactly. Relatively loose stopping criteria are sufficient to achieve fast convergence. In addition, suppose at some iteration $k$, the matrix $(H + \sigma_k M)$ is not positive semidefinite. The preconditioned conjugate gradient algorithm can still be applied as long as it is terminated once the indefiniteness is detected. The vector $v_k$ computed thus far can still be used to expand the subspace. Alternatively, an algorithm that does not require $H + \sigma M$ to be positive definite, such as MINRES or MINRES-QLP, could be used instead of preconditioned conjugate gradient.

At the same time, if the correction equations are sufficiently well conditioned, in the sense that preconditioned conjugate-gradient can be allowed to run to completion, then the algorithm can exhibit rapid convergence, as shown by the following theorem.

**Theorem 4.6.1.** *Suppose the correction equation* (4.49) *is solved exactly at each iteration. Then, if the initial vector $x_1$ is sufficiently close to the optimal solution $x^*$, the sequence of iterates $\{x_k\}$ converges s quadratically to $x^*$, and the sequence of Lagrange multipliers $\{\sigma_k\}$ converges to $\sigma^*$.*

*Proof.* Let $x^*$ and $\sigma^*$ be the optimal solution and Lagrange multiplier to the trust-region subproblem given by $H$, $M$, $g$, and $\delta$. Let $x_1$ and $\sigma_1$ be the starting point of the algorithm, with $r_1 = -g - (H - \sigma_1 M)x_1$ such that $r_1^{\mathrm{T}} x_1 = 0$. By Theorem 4.1.1, $(H + \sigma^* M)x^* + g = 0$. Then $x^*$ can be written as $x^* = x_1 + e_1$, where $e_1$ is the error. Then,

$$\begin{aligned}(H + \sigma_1 M)e_1 &= (H + \sigma_1 M)x^* - (H + \sigma_1 M)x_1 \\ &= (H + \sigma^* M)x^* + (\sigma_1 - \sigma^*)Mx^* - (H + \sigma_1 M)x_1 \\ &= -g - (H + \sigma_1)x_1 + (\sigma_1 - \sigma^*)Mx^* \\ &= r_1 + (\sigma_1 - \sigma^*)Mx^*.\end{aligned} \tag{4.50}$$

Let $v_2$ be the exact solution to the correction equation (4.49), so

$$\left(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1}\right)(H + \sigma_1 M)v_2 = r_1, \quad \text{and} \quad v_2^{\mathrm{T}}Mx_1 = 0.$$

Note that by construction, $(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1})r_1 = r_1$. Now,

$$x^* - (x_1 + v_2) = e_1 - v_2.$$

Thus, for quadratic convergence, it suffices to show that

$$||x^* - (x_1 + v_2)|| = ||e_1 - v_2|| = \mathcal{O}(||e_1||^2).$$

Multiplying (4.50) by the projection operator $(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1})$ and subtracting from the correction equation gives

$$\begin{aligned}\left(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1}\right)&(H + \sigma_1 M)(e_1 - v_2) \\ &= (\sigma_1 - \sigma^*)\left(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1}\right)Mx^* \\ &= (\sigma_1 - \sigma^*)\left(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1}\right)Mx_1 + (\sigma_1 - \sigma^*)\left(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1}\right)Me_1 \\ &= (\sigma_1 - \sigma^*)\left(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1}\right)Me_1.\end{aligned} \tag{4.51}$$

Multiplying (4.50) by $x_1^{\mathrm{T}}$ and using the fact that $r_1^{\mathrm{T}}x_1 = 0$ gives:

$$\sigma_1 - \sigma^* = \frac{x_1^{\mathrm{T}}(H + \sigma_1 M)e_1}{x_1^{\mathrm{T}}Mx^*}. \tag{4.52}$$

Thus,

$$||(\sigma_1 - \sigma^*)\left(I - \frac{Mx_1x_1^{\mathrm{T}}}{x_1^{\mathrm{T}}Mx_1}\right)Me_1|| = \mathcal{O}(||e_1||^2).$$

Assuming that $H + \sigma_1 M$ is nonsingular, the result follows. □

A second advantage Jacobi Davidson trust-region has over SIGLTR is that it can be restarted using the best approximation $x_k$ found thus far without resorting to any block-matrix structure. Once a set number of iterations is reached, the current estimate $x_k$ can be used as the first column of a new basis matrix $V$. This is useful for keeping the memory requirements down to a fixed, reasonable level and the subproblem sufficiently small.

Now, the first several iterations of this method are likely to yield approximations $\sigma_k$ that are far from $\sigma^*$. If $\sigma_k < \sigma^*$, then the matrix $H + \sigma_k M$ may be very poorly conditioned, as $\sigma_k$ may be arbitrarily close to some eigenvalue, making the matrix $H + \sigma_k M$ nearly singular. It is efficient to begin with a low-rank basis matrix $V$ that is simple to compute and yields a reasonable approximation of $\sigma^*$. Thus,

for the first $\ell$ iterations, the correction equation (4.49) is modified, replacing $\sigma_k$ with some $\widehat{\sigma}$ chosen beforehand such that $H + \widehat{\sigma}M$ is positive definite. The larger the value of $\widehat{\sigma}$, the lower the condition number of $H + \widehat{\sigma}M$, making the first few iterations require fewer conjugate-gradient iterations. This allows the method to build up an initial subspace and helps to avoid any iterations where the preconditioned conjugate-gradient method requires too many iterations to be feasible. This is particularly useful if MINRES is used to solve (4.49), as most off-the-shelf implementations of MINRES do not track whether indefiniteness has been detected.

The Jacobi-Davidson trust-region method will converge in at most $n$ iterations, where $n$ is the dimension of the problem. However, convergence is typically achieved significantly faster. If $\ell$ is chosen large enough to avoid any ill-conditioned instances of the correction equation, the algorithm typically requires just a few more iterations. The algorithm may even converge before $\ell$ iterations if the trust-region problem is well-conditioned. Furthermore, any preconditioner of the form $H + \sigma_k M$ may be used to solve the $k$-th correction equation, giving the algorithm quite a bit of flexibility. The preconditioner need not be updated at each iteration. It can instead be reused at each iteration, particularly when $|\sigma_k - \sigma_{k+1}|$ is small.

To accommodate the hard case, including a randomly initialized vector $u$ in the initial matrix $V$ suffices. When restarting, the new matrix $V$ should include a column consisting of the best approximation of the left-most eigenvector.

Overall, the Jacobi-Davison trust-region method behaves quite well in practice. However, it tends to struggle with particularly ill-conditioned problems. The principal advantages of this method over SIGLTR are that any preconditioner can be used and that the linear systems solved at each step need not be solved exactly. Furthermore, as the value of $\sigma$ is updated in the correction equation at each iteration, this method often requires fewer iterations than SIGLTR. However, considerably more computation is needed to compute each basis vector. Therefore, SIGLTR is still recommended if factoring one matrix of the form $H + \sigma M$ is not prohibitively expensive. If, on the other hand, factoring $H + \sigma M$ constitutes the vast majority of the run time of SIGLTR, then the Jacobi-Davidson trust-region method is preferred. Of course, the algorithm presented here is only one potential variant. Other methods that either solve the correction equation using different techniques or solve approximations of the correction equation could be implemented and found to have differing degrees of effectiveness.

The Jacobi-Davidson trust-region algorithm is presented in Algorithm 4.3.

**Algorithm 4.3.** Jacobi-Davidson Trust-Region Algorithm

Given $H, M \in \mathbb{R}^{n\times n}$, $M \succ 0$, $g \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$ with $x_0^{\mathrm{T}} M x_0 \leq \delta^2$, $\sigma_0 \geq 0$ such that $r_0 = -g - (H + \sigma_0 M)x_0$ satisfies $r_0^{\mathrm{T}} x_0 = 0$, and $\varepsilon > 0$.
Given $\ell > 0$ and $m > 0$.
Given $\widehat{\sigma}$ such that $H + \widehat{\sigma} M \succ 0$.
Returns $x$ and $\sigma$ such that $||g + (H + \sigma M)x||_2^2 \leq \varepsilon$ and $\sqrt{x^{\mathrm{T}} M x} \leq (1 + \varepsilon)\delta$.
$k \leftarrow 0$.
$r_0 \leftarrow -g - (H + \sigma_0 M)x_0$.
**if** $||r_0||_2^2 \leq \varepsilon$ **then**
    exit.
**end if**
$v_0 \leftarrow x_0 / \sqrt{x_0^{\mathrm{T}} M x_0}$
$V \leftarrow \begin{bmatrix} v_0 & u \end{bmatrix}$, where $u$ is a random vector.
$k \leftarrow 0$
**while** $||r_k||_2^2 \leq \varepsilon$ **do**
    **if** $k < m$ **then**
        $u \leftarrow \mathrm{argmin}_{u \neq 0, u \in \mathrm{range}(V)} \frac{u^{\mathrm{T}} H u}{u^{\mathrm{T}} M u}$
        $v_k \leftarrow x_k / \sqrt{x_k^{\mathrm{T}} M x_k}$
        $V \leftarrow \begin{bmatrix} v_k u \end{bmatrix}$
    **end if**
    **if** $k \leq \ell$ and $\sigma_k \leq \widehat{\sigma}$ **then**
        $\bar{\sigma} \leftarrow \widehat{\sigma}$
    **else**
        $\bar{\sigma} \leftarrow \sigma_k$
    **end if**
    Solve $(I - \frac{M x_k x_k^{\mathrm{T}}}{x_k^{\mathrm{T}} M x_k})(H + \bar{\sigma} M)(I - \frac{x_k x_k^{\mathrm{T}} M}{x_k^{\mathrm{T}} M x_k}) v_{i+k} = r_k, \quad v_{i+k}^{\mathrm{T}} M x_k = 0$
    $M$-othonormalize $v_{k+1}$ against $V$
    $V \leftarrow \begin{bmatrix} V & v_{k+1} \end{bmatrix}$
    Solve $\min_{y \in \mathbb{R}^{k+1}} \{\frac{1}{2} y^{\mathrm{T}} V^{\mathrm{T}} H V y + g^{\mathrm{T}} V^{\mathrm{T}} y : y^{\mathrm{T}} V^{\mathrm{T}} M V y \leq \delta^2\}$
    $x_{k+1} \leftarrow V y$
    $r_{k+1} \leftarrow -g - (H + \sigma_k M) x_{k+1}$
**end while**

Recall from Section 2.5 the convergence of a trust-region method requires that the computed solution yields a value of the objective function that is less than a fixed fraction of the value of the objective function for some steepest-descent direction. Thus, in order to guarantee convergence, it suffices to choose an initial vector $x_0$ that is the steepest-descent direction in some norm. Since the subspace is expanded at each iteration, the objective value is guaranteed to decrease until restarting. At a restart, the new starting vector is the best approximation thus far, so the value of the objective function cannot increase. Therefore, choosing $x_0 = -\alpha g$ for some scalar $\alpha$ is sufficient convergence of an outer trust-region method using the Jacobi-Davidson trust-region algorithm to solve the trust-region subproblem.

## 4.7 A Locally-Optimal Preconditioned Conjugate-Gradient Trust-Region Algorithm

The main aspect that both SIGLTR and the JDTR have in common is that they work with a sequence of expanding subspaces, i.e., the subspace on which the trust-region subproblem is solved is extended by at least one vector per iteration. As both of these methods are designed to require significantly fewer iterations than GLTR, the subproblem can be expected to remain reasonably low dimensional. Thus each method requires reasonably few iterations. The trade-off is that computing each subsequent basis vector comes at a significant computational effort in each technique. As a result, there may be some problems in which neither of these methods is feasible. The overwhelming majority of optimization algorithms for solving arbitrary problems, such as gradient descent or nonlinear conjugate gradient, operate by computing a new search direction at each iteration, and discarding all previous search directions, in direct contrast to the methods described thus far. Thus, to solve problems in which computing the new search direction in each of the previous methods is infeasible, it stands to reason that a method requiring more iterations, but less computation per iteration, will offer some advantages. The preconditioned conjugate-gradient algorithm for solving linear systems of the form $Hx = b$, where $H$ is positive definite, motivates such a method. The preconditioned conjugate-gradient algorithm can be seen in Algorithm 1.1.

The preconditioned conjugate gradient is derived as a method for minimizing the quadratic function

$$q(x) = \frac{1}{2}x^{\mathrm{T}}Hx + g^{\mathrm{T}}x,$$

where $H$ is symmetric positive definite. An instance of the trust-region problem in which $\sigma = 0$ is thus equivalent to this unconstrained minimization problem and can be solved with the same algorithm. However, the preconditioned conjugate-gradient algorithm cannot be applied to any instance of the trust-region subproblem with $\sigma > 0$. As $\sigma$ is not known, it also cannot be used to solve the system $(H + \sigma M)x = -g$, or equivalently the problem

$$\min_{x \in \mathbb{R}^n} q(x) + \frac{\sigma}{2}(x^{\mathrm{T}}Mx - \delta^2).$$

However, techniques employed by this algorithm can still be used.

In line 11 of Algorithm 1.1, the scalar $\alpha$ is determined by performing an exact line search along the direction $p_k$. As seen in the review of Lanczos-CG, this is equivalent to finding the optimal solution $x_k$ restricted to the subspace $\mathrm{span}\{p_0, \ldots, p_k\}$. In the shifted and inverted GLTR method, this property

is preserved, with the trade-offs of being unable to use an arbitrary preconditioner and needing to either store or regenerate every search vector once the termination criteria are satisfied. In this method, the opposite trade-off is made.

Consider line 12 of the conjugate-gradient algorithm, $x \leftarrow x + \alpha_k p_k$. This is equivalent to performing the operation

$$x \leftarrow \underset{v=x+\alpha p_k, \alpha \in \mathbb{R}}{\operatorname{argmin}} g^{\mathrm{T}} v + \frac{1}{2} v^{\mathrm{T}} H v,$$

which in turn is equivalent to

$$x \leftarrow \underset{v \in \operatorname{span}\{x, p_k\}}{\operatorname{argmin}} g^{\mathrm{T}} v + \frac{1}{2} v^{\mathrm{T}} H v.$$

By line 18, $p_k = z_k + \beta_k p_{k-1}$. Therefore, line 12 is, in exact precision, equivalent to the locally optimal update

$$x \leftarrow \underset{v \in \operatorname{span}\{x, z_k, p_{k-1}\}}{\operatorname{argmin}} g^{\mathrm{T}} v + \frac{1}{2} v^{\mathrm{T}} H v.$$

In the case of the trust-region algorithm, this three-term recursion is modified to suit the structure of the trust-region subproblem. In what follows, $p_{k-1}$ denotes the previous search direction, $r_k$ denotes the current residual $r_k = -g - (H + \sigma_k M) x_k$, where $\sigma_k$ is the current approximation of $\sigma$, and $z_k$ denotes the solution to $B z_k = r_k$, for some positive definite preconditioner $B$. The update step becomes

$$\begin{aligned} x \leftarrow & \underset{v}{\operatorname{argmin}}\ g^{\mathrm{T}} v + \frac{1}{2} x^{\mathrm{T}} H x \\ & \text{subject to} \quad v \in \operatorname{span}\{x, z_k, p_{k-1}\}, \\ & \qquad\qquad ||v|| \leq \delta, \end{aligned} \tag{4.53}$$

and $\sigma_k$ is updated with the value of $\sigma$ from the above low-dimensional trust-region subproblem. The low-dimensional problem can be solved quickly using the Moré-Sorensen algorithm. Therefore, the new vector is a linear combination of the previous estimate, the preconditioned residual, and the previous search direction, i.e.,

$$x_{k+1} = \gamma_k x_k + \alpha_k z_k + \beta_k p_{k-1},$$

where $\gamma_k$, $\alpha_k$, and $\beta_k$, are determined by solving the low dimensional subproblem. The vector $p_k$ is intuitively set to

$$p_k = x_{k+1} - x_k.$$

Observe this update guarantees that $x_k \in \operatorname{span}\{x_{k+1}, p_k\}$. Thus, the previous approximation lies within the subsequent three-dimensional subspace. The basis $\{x_k, z_k, p_{k-1}\}$ is preferred over the equivalent basis

$\{x_k, z_k, x_{k-1}\}$ simply because, as the algorithm converges, $x_k$ and $x_{k-1}$ will only differ slightly. The matrix $\begin{bmatrix} x_k & z_k & x_{k-1} \end{bmatrix}$ will have nearly linearly dependent columns. The new residual is given by

$$\begin{aligned} r_{k+1} &= -g - (H + \sigma_{k+1}M)x_{k+1} \\ &= -g - (H + \sigma_{k+1}M)(x_k + p_k) \\ &= -g - (H + \sigma_k M + (\sigma_{k+1} - \sigma_k)M)(x_k + p_k) \\ &= -g - (H + \sigma_k M)x_k - (H + \sigma_k M)p_k + (\sigma_k - \sigma_{k+1})M(x_k + p_k) \\ &= r_k - (H + \sigma_{k+1}M)p_k + (\sigma_k - \sigma_{k+1})Mx_k \end{aligned} \tag{4.54}$$

Notice that this update differs from the residual update in the conjugate-gradient algorithm in that an extra vector $Mx_k$ is needed. This vector is also used to compute the scalars $\gamma_k, \alpha_k$, and $\beta_{k-1}$. In the preconditioned conjugate-gradient algorithm, the subproblem is solved via an exact line search, which can be calculated explicitly without resorting to any iterative algorithm. In the trust-region case, this property is discarded, and the subproblem is solved via an iterative method. However, it is well known that in floating-point arithmetic, the $H$-conjugacy of the search directions is lost as the algorithm proceeds. In the trust-region case, this implies that the vectors $\{x_k, z_k, p_{k-1}\}$ will eventually become nearly linearly dependent, creating difficulties for the iterative algorithm to solve the subproblem. To mitigate this effect, the Gram-Schmidt biorthogonalization process is used at each iteration to force the search directions to be $M$-orthogonal. In addition, column pivoting can be included to detect better when a breakdown has occurred. This gives enough to form an initial locally-optimal preconditioned conjugate-gradient trust-region algorithm (LOPCGTR).

**Algorithm 4.4.** Locally-Optimal Preconditioned Conjugate-Gradient Trust-Region Algorithm

Given $H, M \in \mathbb{R}^{n\times n}$, $M \succ 0$, $B \in \mathbb{R}^{n\times n}$ such that $Bz = r$ is simple to compute and $B \succ 0$, $x_0 \in \mathbb{R}^n$, $\sigma_0 \geq 0$, $g \in \mathbb{R}^n$, $\varepsilon > 0$.
Returns $x$ and $\sigma$ such that $||g + (H + \sigma M)x||_2^2 \leq \varepsilon$ and $\sqrt{x^{\mathrm{T}} M x} \leq (1 + \varepsilon)\delta$.
$k \leftarrow 0$.
$r_0 \leftarrow -g - (H + \sigma_0 M)x_0$.
**if** $||r_0||_2^2 \leq \varepsilon$ **then**
    exit.
**end if**
Solve $Bz_0 = r_0$
$p_{-1} \leftarrow 0$
**while** Not Converged **do**
    $Z \leftarrow \begin{bmatrix} x_k & z_k & p_{k-1} \end{bmatrix}$
    $(MZ) \leftarrow MZ$
    $(Z, (MZ), R, r) \leftarrow qr(Z, (MZ))$
    $(HZ) \leftarrow HZ$
    $(ZHZ) \leftarrow Z^{\mathrm{T}}(HZ)$
    $(ZMZ) \leftarrow Z^{\mathrm{T}}(MZ)$.
    $(Zg) \leftarrow Z^{\mathrm{T}} g$
    $y \leftarrow \operatorname{argmin}_{y\in\mathbb{R}^r} \{\frac{1}{2} y^{\mathrm{T}}(ZHZ)y + (Zg)^{\mathrm{T}} y \ : \ y^{\mathrm{T}}(ZMZ)y \leq \delta^2\}$
    $p_k \leftarrow (HZ)y - x_k$
    $x_{k+1} \leftarrow x_k + p_k$
    $\sigma_{k+1} \leftarrow$ dual solution
    $r_{k+1} \leftarrow r_k - (H + \sigma_{k+1} M)p_k + (\sigma_k - \sigma_{k+1})Mx_k$
    **if** $r_{k+1}$ is sufficiently small **then**
        exit
    **end if**
    Solve $Bz_{k+1} = r_{k+1}$
**end while**

The operation $(Z, (MZ), R, r) \leftarrow qr(Z, (MZ))$ refers to the action of biorthonormalizing the columns of the matrices $Z$ and $(MZ)$, by left multiplying by the matrix $R$, where $r$ denotes the numerical rank of the result.

This method could be modified to include a larger history of previous search directions so that each iteration solves a subproblem over the subspace $\operatorname{span}\{x_k, z_k, p_{k-1}, \ldots, p_{k-\ell}\}$ for some $\ell > 0$. For the first $\ell$ iterations, the iterates would be, in exact arithmetic, equivalent to the first $\ell$ iterations of shifted and inverted GLTR, assuming the same preconditioner is used. However, the cost of biorthonormalizing the subspace at each iteration grows rapidly with $\ell$. For this reason, the subspace should be limited to one previous search direction.

It is worth noting that this is only one potential implementation of a conjugate-gradient style algorithm for solving the trust-region subproblem. For example, instead of solving the low dimensional subproblem exactly, the update of $x$ could be $x \leftarrow x + \alpha_k p_k$, where $\alpha_k$ would be computed via an exact line search. Due to the simple structure of the trust-region problem, an exact line search in any direction can

be calculated explicitly. The update to the search direction could similarly be replaced by $p_k \leftarrow z_k + \beta_k p_k$, where $\beta_k$ would be computed by any of the numerous expressions used in the nonlinear conjugate-gradient method. However, the trust-region problem has a convenient structure that makes finding the locally optimal update based on the three vectors $x_k$, $z_k$, and $p_{k-1}$ relatively straightforward. Therefore this version is preferred.

As previously mentioned, if the solution $x^*$ to the trust-region problem is an unconstrained minimizer with $||x|| < \delta$, then this algorithm reverts to the standard conjugate-gradient algorithm for solving $Hx = -g$. With this in mind, the algorithm could be broken up into two phases: an initial conjugate-gradient phase that runs until either the trust-region constraint is violated or indefiniteness of $H$ is detected, and then a second phase that runs the locally-optimal preconditioned conjugate-gradient trust-region algorithm.

Now, consider the case where the trust-region subproblem is an instance of the hard case and $x_0 = 0$. Then clearly, this algorithm will not converge to the true solution, as no search directions will ever be generated with a nonzero component in the direction of the leftmost eigenvector $u_n$. Thus, to accommodate convergence to the solution in the hard case from $x_0 = 0$, a secondary step approximating $u = u_n$ must be included. Before updating any vectors, a second subproblem

$$u_{k+1} = \underset{u \in \text{span}\{x_k, u_k, z_k, p_{k-1}\}}{\text{argmin}} \frac{u^{\mathrm{T}} H u}{u^{\mathrm{T}} M u}$$

is solved. The initial choice of $u_0$ must be randomly initialized. Otherwise, the leftmost eigenvector may not be found. This ensures that no components of the leftmost eigenvector are lost as the iterations proceed, yielding a reasonable approximation of the leftmost eigenvector by the time convergence is reached. This is sufficient to formulate a method that converges even in the hard case.

**Algorithm 4.5.** Locally-Optimal Preconditioned Conjugate-Gradient Trust-Region Algorithm V2

Given $H, M \in \mathbb{R}^{n\times n}$, $M \succ 0$, $B \in \mathbb{R}^{n\times n}$ such that $Mz = r$ is simple to compute and $B \succ 0$, $\sigma_0 \geq 0$, $x_0, u_0 \in \mathbb{R}^n$, $g \in \mathbb{R}^n$, $\varepsilon > 0$.
Returns $x$ and $\sigma$ such that $||g + (H + \sigma M)x||_2^2 \leq \varepsilon$ and $\sqrt{x^{\mathrm{T}}Mx} \leq (1+\varepsilon)\delta$.
$r_0 \leftarrow -g - Hx_0$.
**if** $||r_0||_2^2 \leq \varepsilon$ **then**
    exit.
**end if**
Solve $Bz_0 = r_0$
$p_{-1} \leftarrow 0$
**while** Not Converged **do**
    $Z \leftarrow \begin{bmatrix} x_k & u_k & z_k & p_{k-1} \end{bmatrix}$
    $(MZ) \leftarrow MZ$
    $(Z, (MZ), R, r) \leftarrow qr(Z, MZ)$
    $(HZ) \leftarrow HZ$
    $(ZHZ) \leftarrow Z^{\mathrm{T}}(HZ)$
    $(ZMZ) \leftarrow Z^{\mathrm{T}}(MZ)$.
    $(Zg) \leftarrow Z^{\mathrm{T}}g$
    $y \leftarrow \operatorname{argmin}_{y\in\mathbb{R}^r}\{\frac{1}{2}y^{\mathrm{T}}(ZHZ)y + (Zg)^{\mathrm{T}}y \ : \ y^{\mathrm{T}}(ZMZ)y \leq \delta^2\}$
    $q \leftarrow \operatorname{argmin}_{q\in\mathbb{R}^r} \frac{q^{\mathrm{T}}(ZHZ)q}{q^{\mathrm{T}}(ZMZ)q}$
    $p_k \leftarrow Zy - x_k$
    $x_{k+1} \leftarrow x_k + p_k$
    $\sigma_{k+1} \leftarrow$ dual solution
    $u_{k+1} \leftarrow Zq$
    $r_{k+1} \leftarrow r_k - (H + \sigma_{k+1}M)p_k + (\sigma_k - \sigma_{k+1})Mx_k$
    **if** $r_{k+1}$ is sufficiently small **then**
        exit
    **end if**
    Solve $Bz_{k+1} = r_{k+1}$
**end while**

The preconditioner $B$ is chosen so that $B \approx H + \sigma^* M$, where $\sigma^*$ is the optimal Lagrange multiplier. Unfortunately, $\sigma^*$ is not known ahead of time, so instead, $M$ is chosen such that $B \approx H + \widehat{\sigma}M$, where $\widehat{\sigma}$ is some approximation of $\sigma^*$. In most implementations of preconditioned conjugate-gradient, a restarting scheme is utilized to help mitigate the effects of the numerical instability of the Gram-Schmidt process. The LOPCGTR algorithm can be restarted similarly. Say the algorithm is restarted after $m$ iterations. Before the algorithm is resumed, the preconditioner $B$ may be updated using the current estimate of $\sigma$ so that $B \approx H + \sigma_m M$. This can significantly reduce the number of iterations until convergence and helps to ensure that not too many restarts are required.

In the standard conjugate-gradient algorithm, the residual $r_k = b - Hx_k$ is the gradient of the quadratic objective function $\frac{1}{2}x^{\mathrm{T}}Hx - b^{\mathrm{T}}x$. Therefore, the conjugate-gradient method is guaranteed to converge at least as fast as gradient descent and is guaranteed to converge. However, in the LOPCGTR algorithm, $r_k = -g - (H + \sigma_k M)x_k$ is not the gradient of any obvious function. Instead, it is the derivative

of the Lagrangian function $L$ with respect to the primal variables $x$. To better motivate convergence, it would help to show that the residual vector $r_k$ is colinear with the gradient of some unconstrained objective function that shares a global minimum $x^*$ with the trust-region problem.

Consider the case where the solution $x$ satisfies the trust-region constraint so that $||x||_M = \delta$. Then the value of the objective function $q(x)$ is equivalent to the value of the function

$$w(x) = \delta^2 \frac{\frac{1}{2}x^{\mathrm{T}}Hx + \dfrac{||x||_M}{\delta}g^{\mathrm{T}}x}{x^{\mathrm{T}}Mx}.$$

This function is continuous everywhere except at $x = 0$. Furthermore, for any $x \neq 0$, $w(\alpha x) = \alpha w(x)$ for all $\alpha > 0$. Thus, if $x$ is a constrained global minimizer of the trust-region subproblem, then $\alpha x$ is a global minimizer of $w$ for all $\alpha > 0$. Note that for any $\alpha > 0$, $\alpha x$ is not an isolated minimizer of $w$. Now, $w$ and $q$ agree on the trust-region boundary but not on the interior of the trust region. Let

$$f(x) = \delta^2 \frac{\frac{1}{2}x^{\mathrm{T}}Hx + \dfrac{\max(||x||_M, \delta)}{\delta}g^{\mathrm{T}}x}{\max(x^{\mathrm{T}}Mx, \delta^2)}.$$

Then $f(x) = q(x)$ for all $x$ such that $||x||_M \leq \delta$, and if $||x||_M = \delta$, $f(\alpha x) = q(x)$ for all $\alpha \geq 1$. Thus, if $x$ is either an unconstrained or constrained global minimizer of the trust-region problem, then $x$ is an unconstrained global minimizer of $f$. On the other hand, if $x$ is an unconstrained minimizer of $f$, then $\bar{x} = (\delta / \max(\delta, ||x||_M))x$ is a minimizer of the trust-region problem. Thus, techniques for solving unconstrained problems (such as the nonlinear conjugate-gradient method) can be used to solve the trust-region problem. The gradient of $f$ is clearly discontinuous on $\{x : ||x||_M = \delta\}$. However, it will be shown that this does not present any difficulties. First, on the interior of the trust-region $\{x : ||x||_M < \delta\}$, $f(x) = q(x)$, so

$$\nabla f(x) = g + Hx.$$

On the set $\{x : ||x||_M > \delta\}$, $f(x) = w(x)$, so $\nabla f(x) = \nabla w(x)$, and

$$\nabla w(x) = \frac{\delta^2}{x^{\mathrm{T}}Mx}\left(\frac{||x||_M}{\delta}g + Hx - \left(\frac{\dfrac{||x||_M}{\delta}g^{\mathrm{T}}x + x^{\mathrm{T}}Hx}{x^{\mathrm{T}}Mx}\right)Mx\right).$$

Let $\{x_n\} \subset \{x : ||x||_M > \delta\}$ be a sequence outside the trust-region that converges to a point $\widehat{x}$ on the trust-region boundary. Then

$$\lim_{x_n \to \widehat{x}} \nabla f(x) = g + Hx + \bar{\sigma}(x)Mx,$$

where

$$\bar{\sigma}(x) = -\frac{\frac{||x||_M}{\delta} g^{\mathrm{T}} x + x^{\mathrm{T}} H x}{x^{\mathrm{T}} M x}.$$

If

$$\tilde{\sigma}(x) = \begin{cases} 0 & \text{if } ||x||_M < \delta \\ \bar{\sigma}(x) & \text{if } ||x||_M \geq \delta, \end{cases}$$

then

$$\nabla f(x) = \frac{\delta^2}{\max(x^{\mathrm{T}} M x, \delta^2)} \left( \frac{\max(||x||_M, \delta)}{\delta} g + Hx + \tilde{\sigma}(x) M x \right)$$

everywhere but the boundary of the trust region. Let

$$h(x) = \frac{\delta^2}{\max(x^{\mathrm{T}} M x, \delta^2)} \left( \frac{\max(||x||_M, \delta)}{\delta} g + Hx + \tilde{\sigma}(x) M x \right),$$

with domain $\mathbb{R}^n$. Suppose that $x$ is a minimizer of $f$ satisfying $||x||_M = \delta$. Then $x$ is a constrained minimizer of the trust-region problem. Therefore, a $\sigma > 0$ exists such that $g + Hx + \sigma M x = 0$. By left multiplying by $x$, it holds that

$$\sigma = -\frac{g^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} H x}{x^{\mathrm{T}} M x} = -\frac{\frac{||x||_M}{\delta} g^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} H x}{x^{\mathrm{T}} M x} = \tilde{\sigma}(x).$$

Thus, if $x$ is a minimizer of $f$, then $h(x) = 0$, even where $\nabla f(x)$ is undefined. The residual vector $r_k = -g - (H + \sigma_k M) x_k$ used in the LOPCGTR algorithm is equivalent to $-g - (H + \tilde{\sigma}(x_k) M) x_k$ when $x$ is restricted to the trust-region, and is colinear with $h(x)$. It is worth mentioning that, in the constrained case, $\sigma^*$ does not necessarily maximize the function $\tilde{\sigma}(x)$. Note that the locally optimal update step guarantees each iterate $x_k$ lies within the trust region. Therefore, the LOPCGTR algorithm can be expected to converge at least as fast as the nonlinear conjugate-gradient algorithm applied to the function $f(x)$, and is guaranteed to converge.

Overall, the trust-region conjugate-gradient algorithm performs in a very predictable manner. Convergence requires more iterations than in the shifted and inverted SIGLTR algorithm when $B = H + \mu M$, where $\mu$ is the shift parameter. This is because subsequent iterations discard previous information. However, the subproblem being solved is limited to four dimensions, unlike SIGLTR, in which the subproblem adds a new search direction at each iteration without discarding any previous information. Furthermore, the computation required to compute each new search direction only requires the solution of the preconditioning equation $Bz_k = r_k$ instead of the significantly more costly $(H + \mu M)u = v$. In addition, the conjugate-

gradient trust region method is considerably more flexible in choosing a preconditioner. Any positive definite matrix $B$ can be used, whereas SIGLTR is restricted to preconditioners of the form $B = H + \mu M$. Overall, SIGLTR is still the preferred method if one factorization of $H + \mu M$ is not prohibitively expensive.

Experiments also indicate that this conjugate-gradient style algorithm performs much more favorably than the Jacobi-Davidson style algorithm in practically all cases. At best, the Jacobi-Davidson style algorithm required a comparable number of conjugate-gradient iterations while requiring the additional overhead of orthogonalizing the update vector against every computed vector thus far. The only true advantage the Jacobi-Davidson trust-region method maintains is that the correction equation can be solved via any means possible, simplifying the method if the objective and constraint matrices have a particular structure.

Recall from Section 2.5 the convergence of a trust-region method requires that the computed solution yields a value of the objective function that is less than a fixed fraction of the value of the objective function for some steepest-descent direction. If the locally-optimal preconditioned conjugate-gradient algorithm is not warm started and used a preconditioner $B$, the first search space is a one-dimensional subspace spanned by $z_1 = B^{-1}r_1 = -B^{-1}g$. As $B$ is chosen to be positive definite, $z_1$ is the steepest-descent direction in the $B$ norm. Furthermore, the current subspace $\text{span}\{x_k, r_k, p_{k-1}\}$ always contains $x_k$ and $x_{k-1}$, so the value of the objective function can only decrease at each iteration. Thus, the locally-optimal preconditioned conjugate-gradient algorithm guarantees a sufficient reduction of the quadratic objective function after 1 iteration.

## 4.8 Doubly-Augmented Trust-Region Problems

In the trust-region method for unconstrained optimization, the trust-region subproblem is typically of the form

$$\min_{d \in \mathbb{R}^n} \; d^{\mathrm{T}} g + \frac{1}{2} d^{\mathrm{T}} H d$$

$$\text{subject to} \quad ||d|| \leq \delta,$$

where $g$ is the gradient of the objective function $f$, and $H$ is the (potentially indefinite) Hessian of $f$. Depending on the problem, Moré-Sorensen, SIGLTR, or LOPCGTR can reliably solve the trust-region subproblem to any arbitrary level of accuracy. In the constrained case, as shall be seen in later chapters,

the trust-region method can be applied with trust-region subproblems of the form

$$\begin{aligned} \min_{d_x \in \mathbb{R}^n, d_y \in \mathbb{R}^m} \quad & \begin{pmatrix} g_x^{\mathrm{T}} & g_y^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} d_x \\ -d_y \end{pmatrix} + \frac{1}{2} \begin{pmatrix} d_x^{\mathrm{T}} & -d_y^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} H + 2J^{\mathrm{T}} D^{-1} J & -J^{\mathrm{T}} \\ -J & D \end{pmatrix} \begin{pmatrix} d_x \\ -d_y \end{pmatrix} \\ \text{subject to} \quad & \begin{pmatrix} d_x^{\mathrm{T}} & -d_y^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} B_{1,1} & B_{2,1}^{\mathrm{T}} \\ B_{2,1} & B_{2,2} \end{pmatrix} \begin{pmatrix} d_x \\ -d_y \end{pmatrix} \leq \delta^2, \end{aligned} \tag{4.55}$$

where $D$ is a diagonal and positive definite matrix. Let

$$H^M = \begin{pmatrix} H + 2J^{\mathrm{T}} D^{-1} J & -J^{\mathrm{T}} \\ -J & D \end{pmatrix}, \quad \text{and} \quad B = \begin{pmatrix} B_{1,1} & B_{2,1}^{\mathrm{T}} \\ B_{2,1} & B_{2,2} \end{pmatrix}.$$

The superscript on $H^M$ denotes that $H$ is the Hessian matrix of some merit function. Due to the $2J^{\mathrm{T}} D^{-1} J$ term in the upper diagonal block, the matrix $B$ is referred to as *doubly augmented*. The doubly-augmented structure of $H^M$ makes this problem considerably more difficult to solve than the general sparse case. If $J$ is sparse, then even performing the matrix triple product $J^{\mathrm{T}} D^{-1} J$ may be prohibitively expensive. Moreover, if $J$ has a single dense row, the resulting matrix $H^M$ will have a dense upper left block.

Additionally, the objective matrix $H^M$ is extremely ill-conditioned in the Euclidean norm, particularly when the elements of $D$ approach either 0 or infinity. Thus, special consideration must be given to any implementations of the previous algorithms for problems of this form. It is also crucial that the matrix $B$ be chosen carefully, so whatever special treatment is given to the matrix $H^M$ can be given to matrices $H^M + \sigma B$. For this reason, two special cases of the matrix $B$ are considered:

$$B^{(1)} = \begin{pmatrix} I & 0 \\ 0 & D \end{pmatrix}, \quad \text{and} \quad B^{(2)} = \begin{pmatrix} I + 2J^{\mathrm{T}} D^{-1} J & -J^{\mathrm{T}} \\ -J & D \end{pmatrix}, \tag{4.56}$$

both of which are positive definite.

*Proof.* $B^{(1)}$ is clearly positive definite given that $D$ is positive definite. Let $\widehat{R}$ denote the matrix

$$\widehat{R} = \begin{pmatrix} I & J^{\mathrm{T}} D^{-1} \\ 0 & I \end{pmatrix}.$$

Then

$$\widehat{R}B^{(2)}\widehat{R}^{\mathrm{T}} = \begin{pmatrix} I + J^{\mathrm{T}}D^{-1}J & 0 \\ 0 & D \end{pmatrix}.$$

Note that $R$ is nonsingular, with

$$\widehat{R}^{-1} = \begin{pmatrix} I & -J^{\mathrm{T}}D^{-1} \\ 0 & I \end{pmatrix}.$$

The identity matrix is positive definite, and the matrix $J^{\mathrm{T}}D^{-1}J$ is positive semidefinite. Thus, the Schur complement matrix $I + J^{\mathrm{T}}D^{-1}J$ is also positive definite. Therefore, $\widehat{R}B^{(2)}\widehat{R}^{\mathrm{T}}$ is positive definite. For any $v \in \mathbb{R}^n$, $v^{\mathrm{T}}B^{(2)}v = v^{\mathrm{T}}\widehat{R}^{-1}(\widehat{R}B^{(2)}\widehat{R}^{\mathrm{T}})(\widehat{R}^{\mathrm{T}})^{-1}v \geq 0$, with equality if $v = 0$. Thus, $B^{(2)}$ is positive definite. □

For both the Moré Sorensen and SIGLTR algorithms, inertia-revealing factors of $H^M + \sigma M$ are required, i.e., factorizations that also provide the integer triple $\mathrm{In}(H^M + \sigma M) = (n_+, n_-, n_0)$, where $n_+$ is the number of positive eigenvalues, $n_-$ is the number of negative eigenvalues, and $n_0$ is the number of 0 eigenvalues. The complete inertia is not strictly necessary, as a Cholesky decomposition can suffice to determine if a matrix is positive definite. That being said, it is essential to factor matrices of this form without computing the matrix triple product, and use said factors in determining whether the matrix is positive definite. Let

$$R = \begin{pmatrix} I & 2J^{\mathrm{T}}D^{-1} \\ 0 & -I \end{pmatrix}. \tag{4.57}$$

Observe that $R^2 = I$. Define the following matrices:

$$K = RH^M = \begin{pmatrix} H & J^{\mathrm{T}} \\ J & -D \end{pmatrix},$$

$$T^{(1)} = RB^{(1)} = \begin{pmatrix} I & 2J^{\mathrm{T}} \\ 0 & -D \end{pmatrix}, \text{ and}$$

$$T^{(2)} = RB^{(2)} = \begin{pmatrix} I & J^{\mathrm{T}} \\ J & -D \end{pmatrix}.$$

It then holds that $H^M + \sigma B^{(i)} = R(K + \sigma T^{(i)})$, and $(H^M + \sigma B^{(i)})^{-1} = (K + \sigma T^{(i)})^{-1}R$ for $i = 1, 2$. Thus,

it is sufficient to factor matrices of the form $K + \sigma T^{(i)}$. Consider the case $i = 1$. Then

$$K + \sigma T^{(1)} = \begin{pmatrix} H + \sigma I & (1 + 2\sigma)J^{\mathrm{T}} \\ J & -(1 + \sigma)D \end{pmatrix}.$$

This matrix is not symmetric. However, the columns can be scaled so that the resulting matrix is symmetric. Let

$$S = \begin{pmatrix} I & 0 \\ 0 & \frac{1}{1+2\sigma} \end{pmatrix}, \quad \text{and} \quad \widehat{K}_\sigma^{(1)} = (K + \sigma T^{(1)})S = \begin{pmatrix} H + \sigma I & J^{\mathrm{T}} \\ J & -\frac{1+\sigma}{1+2\sigma}D \end{pmatrix}. \tag{4.58}$$

Let $\widehat{\sigma} = (1+\sigma)/(1+2\sigma)$. As this matrix is symmetric, an inertia-revealing symmetric indefinite factorization can be applied.

**Theorem 4.8.1.** *$H^{M} + \sigma B^{(1)}$ is positive definite if and only if $\widehat{K}_\sigma^{(1)}$ has exactly $n$ positive and $m$ negative eigenvalues.*

*Proof.*

$$H^{M} + \sigma B^{(1)} = \begin{pmatrix} H + \sigma I + 2J^{\mathrm{T}}D^{-1}J & -J^{\mathrm{T}} \\ -J & (1 + \sigma)D \end{pmatrix}, \tag{4.59}$$

so

$$\begin{aligned} \begin{pmatrix} I & \frac{1}{1+\sigma}J^{\mathrm{T}}D^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} H + \sigma I + 2J^{\mathrm{T}}D^{-1}J & -J^{\mathrm{T}} \\ -J & (1 + \sigma)D \end{pmatrix} \begin{pmatrix} I & 0 \\ \frac{1}{1+\sigma}D^{-1}J & I \end{pmatrix} \\ = \begin{pmatrix} H + \sigma I + \frac{1}{\widehat{\sigma}}J^{\mathrm{T}}D^{-1}J & \\ & (1 + \sigma)D \end{pmatrix}. \end{aligned} \tag{4.60}$$

Therefore, $H^{M} + \sigma B^{(1)}$ is positive definite if an only if $H + \sigma I + \frac{1}{\widehat{\sigma}}J^{\mathrm{T}}D^{-1}J$ is positive definite. Conversely,

$$\begin{pmatrix} I & \frac{1}{\widehat{\sigma}}J^{\mathrm{T}}D^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} H + \sigma I & J^{\mathrm{T}} \\ J & -\widehat{\sigma}D \end{pmatrix} \begin{pmatrix} I & 0 \\ \frac{1}{\widehat{\sigma}}D^{-1}J & I \end{pmatrix} = \begin{pmatrix} H + \sigma I + \frac{1}{\widehat{\sigma}}J^{\mathrm{T}}D^{-1}J & \\ & -\widehat{\sigma}D \end{pmatrix},$$

therefore $\mathrm{In}(\widehat{K}_\sigma^{(1)}) = (n, m, 0)$ if and only if $H + \sigma I + \frac{1}{\widehat{\sigma}}J^{\mathrm{T}}D^{-1}J$ is positive definite. The result follows. □

Now, consider the case $i = 2$. Then

$$K + \sigma T^{(2)} = \begin{pmatrix} H + \sigma I & (1 + \sigma)J^{\mathrm{T}} \\ (1 + \sigma)J & -(1 + \sigma)D \end{pmatrix}.$$

This matrix is already symmetric, so no additional transformations are required. The following result shows that the inertia of $H^{M} + \sigma B^{(2)}$ can be inferred from the inertia of $K + \sigma T^{(2)}$.

**Theorem 4.8.2.** *The matrix $H^{M} + \sigma B^{(2)}$ is positive definite if and only if* $\text{In}(K + \sigma T^{(2)}) = (n, m, 0)$.

*Proof.* Consider the transformation

$$\begin{pmatrix} I & J^{\mathrm{T}} D^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} H + \sigma I + 2(1+\sigma) J^{\mathrm{T}} D^{-1} J & -(1+\sigma) J^{\mathrm{T}} \\ -(1+\sigma) J & (1+\sigma) D \end{pmatrix} \begin{pmatrix} I & 0 \\ D^{-1} J & I \end{pmatrix} = \begin{pmatrix} H + \sigma I + (1+\sigma) J^{\mathrm{T}} D^{-1} J & \\ & (1+\sigma) D \end{pmatrix}.$$

Therefore, $H^{M} + \sigma B^{(2)}$ is positive definite if an only if $H + \sigma I + (1+\sigma) J^{\mathrm{T}} D^{-1} J$ is positive definite. Conversely,

$$\begin{pmatrix} I & J^{\mathrm{T}} D^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} H + \sigma I & (1+\sigma) J^{\mathrm{T}} \\ (1+\sigma) J & -(1+\sigma) D \end{pmatrix} \begin{pmatrix} I & 0 \\ D^{-1} J & I \end{pmatrix} = \begin{pmatrix} H + \sigma I + (1+\sigma) J^{\mathrm{T}} D^{-1} J & \\ & -(1+\sigma) D \end{pmatrix}.$$

Therefore, $\text{In}(K + \sigma T^{(2)}) = (n, m, 0)$ if and only if $H + \sigma I + (1+\sigma) J^{\mathrm{T}} D^{-1} J$ is positive definite. The result follows. □

Note that any system involving $H^{M} + \sigma B^{(i)}$ can be solved in terms of $\widehat{K}_{\sigma}^{(1)}$, $K + \sigma T^{(2)}$ and the transformations $R$ and $S$. Thus, any factorizations that need to be performed can be done with an inertia-revealing symmetric indefinite factorization, allowing any of the algorithms discussed in the previous sections for solving the trust-region problem to be applied. This is particularly beneficial for the Moré Sorensen and SIGLTR algorithms, regardless of whether the problem being solved uses $B^{(1)}$ or $B^{(2)}$, as $\widehat{K}_{\sigma}^{(1)}$ and $K + \sigma T^{(2)}$ share the same sparsity pattern.

However, some remarks must be made before applying the standard GLTR and the LOPCGTR algorithms. Recall that the SIGLTR algorithm utilizes a matrix of the form $H^{M} + \mu B^{(i)}$ as a preconditioner in the Lanczos process, whereas GLTR uses $B^{(i)}$. The matrix $B^{(1)}$ is diagonal and thus does not require any factorizations to solve the system $B^{(1)} u = v$. On the other hand, $B^{(2)}$ has the same doubly-augmented structure as $H^{M}$ and cannot be directly factored. Instead, the matrix $T^{(2)}$ can be factored, and the system $B^{(2)} u = v$ can be solved via $B^{(2)} u = R T^{(2)} u = v$. Thus, applying the GLTR algorithm may be

limited to $i = 1$. At the same time, consider the case where the trust-region subproblem given by $H^M$ and $B^{(1)}$ has $\sigma = 0$. Then the GLTR algorithm becomes the preconditioned conjugate-gradient method using the matrix $B^{(1)}$ as a preconditioner. This system is poorly conditioned and thus may take far too many iterations per trust-region subproblem to be considered viable. These factors must be considered before choosing an algorithm.

The LOPCGTR algorithm requires some remarks as well. As in the case of preconditioned conjugate gradient, the algorithm's success largely depends on the choice of the preconditioner. For both $H^M + \sigma B^{(1)}$ and $H^M + \sigma B^{(2)}$, an appropriate choice of the preconditioner is far from obvious due to the presence of the doubly-augmented upper left block. In experiments, preconditioners $N^{(i)}$ were chosen to have the form

$$N^{(1)} = \begin{pmatrix} G + \mu I + 2J^{\mathrm{T}} D^{-1} J & -J^{\mathrm{T}} \\ -J & (1+\mu)D \end{pmatrix}$$

and

$$N^{(2)} = \begin{pmatrix} G + \mu I + 2(1+\mu) J^{\mathrm{T}} D^{-1} J & -(1+\mu) J^{\mathrm{T}} \\ -(1+\mu) J & (1+\mu) D \end{pmatrix},$$

where $G$ denotes the diagonal matrix such that $G_{i,i} = H_{i,i}$. Equations of the form $N^{(i)} u = v$ can be solved using the same transformations $R$ and $T$ and an indefinite matrix with a similar sparsity pattern as $K + \mu T^{(i)}$. In a typical problem, factoring $N^{(i)}$ should take less computation than factoring $H^M + \sigma B^{(i)}$. However, if the dimension of the problem is particularly large, this may still be prohibitively expensive.

The common thread between the application of Moré Sorensen, GLTR, SIGLTR, and LOPCGTR to the doubly-augmented trust-region problem is that particularly large problems may be out of reach for all four methods, especially when at least one factorization of a saddle point matrix is required. What is needed is a method that does not require any sparse factorization techniques but can still use an arbitrary preconditioner to combat any ill-conditioning inherent to the problem. The Jacobi-Davidson trust-region method, while an impractical choice in the general case, can be modified to meet these goals.

### 4.8.1 The Jacobi-Davidson QZ Trust-Region Algorithm

The modifications to the Jacobi-Davidson algorithm for solving the doubly-augmented trust-region problem are inspired by the JDQZ algorithm for solving indefinite generalized eigenvalue problems. See [10] for the details of the JDQZ algorithm. Hence, this algorithm is dubbed the Jacobi-Davidson QZ trust-region algorithm (JDQZTR). This method exploits the relationship between the potentially dense trust-region equations and the sparse indefinite saddle point systems of the previous section. Consider

problem (4.55), with either $B = B^{(1)}$ or $M = B^{(2)}$. By Theorem 4.1.1, the optimality conditions of this problem are the following:

1. $(H^M + \sigma B^{(i)})x + g = 0$,
2. $x^{\mathrm{T}} B^{(i)} x \leq \delta^2$,
3. $\sigma \leq 0$,
4. $\sigma(\delta^2 - x^{\mathrm{T}} B^{(i)} x) = 0$, and
5. $H^M + \sigma B^{(i)} \succeq 0$.

Let $\tilde{g} = Rg$. The first condition can be written as

$$(K + \sigma T^{(i)})x + \tilde{g} = 0.$$

The final condition is equivalent to the condition that the matrix $H^M + \sigma B^{(i)}$ has only positive eigenvalues. If $i = 1$, then this is equivalent to $\widehat{K}_\sigma^{(1)}$ having the correct inertia, and if $i = 2$, $K + \sigma T^{(2)}$ having the correct inertia. However, in the large-scale case, in which inertia-revealing factorizations are not viable, this inertia condition is not particularly useful, as using it would require making commentary on every single eigenvalue of an indefinite matrix. On the other hand, showing that all eigenvalues are positive only requires commentary on the smallest eigenvalue, i.e., if the smallest eigenvalue is positive, then all eigenvalues are positive. This motivates the following result.

**Theorem 4.8.3.** *The matrix $H^M + \sigma B^{(i)}$, for some $\sigma \in \mathbb{R}^n$ and $i \in \{1, 2\}$ is positive semidefinite if and only if the generalized eigenvalue problem $(K + \sigma T^{(i)})x = \lambda T^{(i)}$ has only nonnegative solutions.*

*Proof.* Suppose the matrix $H^M + \sigma B^{(i)}$ has inertia $(n_+, n_-, n_0)$. As $B^{(i)}$ is positive definite, it has a unique positive definite square root. By Sylvester's Law of Inertia, $\mathrm{In}((B^{(i)})^{-1/2}(H^M + \sigma B^{(i)})(B^{(i)})^{-1/2}) = (n_+, n_-, n_0)$. The eigenvalue problem

$$(B^{(i)})^{-1/2}(H^M + \sigma B^{(i)})(B^{(i)})^{-1/2}x = \lambda x$$

is equivalent to

$$(H^M + \sigma B^{(i)})y = \lambda B^{(i)} y,$$

where $y = (B^{(i)})^{-1/2}x$. Right multiplying by the matrix $R$ gives

$$(K + \sigma T^{(i)})y = \lambda T^{(i)}y.$$

The result follows. □

The Jacobi-Davidson method, as presented earlier, seeks to solve the standard trust-region optimality conditions. The goal here is to devise a new method that seeks to solve the following equivalent conditions:

1. $(K + \sigma T^{(i)})x + \tilde{g} = 0$,
2. $x^{\mathrm{T}} B^{(i)} x \leq \delta^2$,
3. $\sigma \geq 0$,
4. $\sigma(\delta^2 - x^{\mathrm{T}} B^{(i)} x) = 0$, and
5. $(K + \sigma T^{(i)}, T^{(i)})$ has no negative eigenvalues.

As $(K + \sigma T^{(i)}, T^{(i)})$ is not a symmetric-definite pencil, one would typically also have to add a condition that all of the eigenvalues of are real $(K + \sigma T^{(i)}, T^{(i)})$ as well. However, the proof of Theorem 4.8.3 also shows that all eigenvalues of $(K + \sigma T^{(i)}, T^{(i)})$ are real.

The main idea of the Jacobi-Davidson method is to build a suitable subspace on which to solve a projected trust-region subproblem. The projected trust-region subproblem in JDTR is derived via a Galerkin condition. However, due to the indefinite nature of the pencil under consideration, a Galerkin condition may no longer be appropriate. Instead, consider a Petrov-Galerkin condition, in which the search space range($V$) is distinct from the test space range($W$) for some basis matrices $V$ and $W$. Let $r$ denote the residual vector $(K + \sigma T^{(i)})x + \tilde{g} = 0$. Given a search space range($V$) and a test space range($W$), the method seeks to solve the following projected conditions:

$$W^{\mathrm{T}}(K + \sigma T^{(i)})Vy + W^{\mathrm{T}}\tilde{g} = 0 \tag{4.61a}$$

$$y^{\mathrm{T}} V^{\mathrm{T}} B^{(i)} V y \leq \delta^2 \tag{4.61b}$$

$$\sigma \geq 0 \tag{4.61c}$$

$$\sigma(\delta^2 - y^{\mathrm{T}} V^{\mathrm{T}} B^{(i)} V y) = 0 \tag{4.61d}$$

$$(W^{\mathrm{T}} K V + \sigma W^{\mathrm{T}} T^{(i)} V, W^{\mathrm{T}} T^{(i)} V) \text{ has no real eigenvalues less than zero.} \tag{4.61e}$$

Projecting the generalized eigenvalue problem onto $V$ and $W$ may introduce complex eigenvalues. Observe that condition 1 above is equivalent to finding a vector $x \in V$ such that

$$(K + \sigma T^{(i)})x + \tilde{g} \perp W.$$

To formulate a practical method out of these conditions, how to construct the subspaces $V$ and $W$ must be understood. Additionally, a strategy is needed for solving the projected subproblem.

Suppose, at iteration $k$, $k$ orthonormal basis vectors for both $k$-dimensional spaces $V_k$ and $W_k$ are known, and some approximate solution $x_k \in V_k$ and $\sigma_k \geq 0$ such that the residual vector $r_k = -(K + \sigma_k T^{(i)})x_k - \tilde{g}$ is orthogonal to $W_k$ has been computed. The next basis vector is found by approximately solving the correction equation

$$\left(I - \frac{z_k z_k^{\mathrm{T}}}{z_k^{\mathrm{T}} z_k}\right)(K + \sigma_k T^{(i)})\left(I - \frac{x_k x_k^{\mathrm{T}}}{x_k^{\mathrm{T}} x_k}\right) v_{k+1} = r_k, \quad v_{k+1}^{\mathrm{T}} x_k = 0. \tag{4.62}$$

The key difference between (4.62) and (4.49) is that the projection operator on the left-hand side differs from the projection operator on the right. Following [10], $z_k$ is chosen to lie in the span of $Kx_K$ and $Tx_k$, i.e., $z_k = \alpha_k K x_k + \beta_k T x_k$ for some scalars $\alpha_k$ and $\beta_k$. The scheme for choosing $\alpha_k$ and $\beta_k$ is postponed until the correction equation is further examined. As in the case of the symmetric definite Jacobi-Davidson algorithm, the convergence is quadratic if the correction equation is solved exactly.

Once $v_{k+1}$ has been found, $V_{k+1}$ is taken to be $[V_k, v_{k+1}]$, and thus a new vector $w_{k+1}$ needs to be found for $W_{k+1}$. Following the construction of $z_k$, $w_{k+1}$ is chosen to lie in the space $\alpha_k K V_{k+1} + \beta_k T V_{k+1}$. A straightforward choice is

$$w_{k+1} = \alpha_k K v_{k+1} + \beta_k T v_{k+1}.$$

Once $v_{k+1}$ and $w_{k+1}$ have been computed, they are orthogonalized against the previous $k$ basis vectors using the modified Gram-Schmidt process. Then, the next projected subproblem is solved.

Some care needs to be taken when solving the projected subproblem. In the symmetric definite case, the projected subproblem was a low dimensional trust-region problem, and thus the Moré-Sorensen algorithm could be applied without any complications. In this case, it is less clear how to formulate a solution. At iteration $k$, let $y_k$ denote a vector in $\mathbb{R}^k$ that satisfies (4.61), and let $\sigma_k$ be the associated multiplier. Then $x_k = V_k y_k$, and $\sigma_k$ will be used to formulate the $k+1$-th correction equation.

Let $\bar{K}_k = W_k^{\mathrm{T}} K V_k$, $\bar{T}_k = W_k^{\mathrm{T}} T V_k$, $\bar{M}_k = V_k^{\mathrm{T}} M V_k$, and $\bar{g}_k = W_k^{\mathrm{T}} \tilde{g}$. Like the Moré-Sorensen algorithm, the goal is to create a sequence of trial multipliers $\{\bar{\sigma}_k\}$ converging to $\sigma_k$. Unfortunately,

there is no direct inertia-revealing factorization for determining if all real eigenvalues of the matrix pencil $(\bar{K}_k + \bar{\sigma}_j \bar{T}_k, \bar{T}_k)$ are nonnegative, given a trial value $\bar{\sigma}_j$. Fortunately, the projected subproblem is low dimensional, so more computationally intensive techniques can be used. Define the matrices $Q_k$, $Z_k$, $S_k$, and $P_k$ as factors in the real, generalized Schur decomposition of $(\bar{K}_k, \bar{T}_k)$, i.e.,

$$Q_k^{\mathrm{T}} \bar{K}_k Z_k = S_k, \quad \text{and} \quad Q_k^{\mathrm{T}} \bar{T}_k Z_k = P_k,$$

where $Q_k, Z_k \in R^k$ are real and orthogonal, $S_k \in \mathbb{R}^k$ is a real, upper quasi-triangular matrix (block upper triangular with blocks of size one or two), and $P_k \in \mathbb{R}^k$ a real, upper triangular matrix with nonnegative values along the diagonal. Such a decomposition always exists. The real eigenvalues of $(\bar{K}_k, \bar{T}_k)$ are given by the ratios of the diagonal entries of $S_k$ and $P_k$, where the real eigenvalues correspond to the $1 \times 1$ blocks of $S_k$. The complex eigenvalues correspond to the $2 \times 2$ blocks. The eigenvalues of $(\bar{K}_k + \bar{\sigma}_j \bar{T}_k, \bar{T}_k)$ can therefore be easily found for any value of $\bar{\sigma}_j$. Let $\lambda_{\min}^{(k)}$ denote the lowest real eigenvalue of the pencil. Then, for any $\sigma > -\lambda_{\min}^{(k)}$, the system $(\bar{K}_k + \sigma \bar{T}_k) y_k = -\bar{g}_k$ has a unique solution. If the Schur decomposition has been found, the entire system can be transformed to use the Schur factors. Let $z_k = Z_k^{\mathrm{T}} y_k$, $h_k = Q_k^{\mathrm{T}} \bar{g}_k$, and $N_k = Z_k^{\mathrm{T}} M Z_k$. Then (4.61) becomes

$$(S_k + \sigma P_k) z_k + h_k = 0 \tag{4.63a}$$

$$z_k^{\mathrm{T}} N_k z_k \leq \delta^2, \tag{4.63b}$$

$$\sigma \geq 0, \tag{4.63c}$$

$$\sigma(\delta^2 - z_k^{\mathrm{T}} N_k z_k) = 0, \text{ and} \tag{4.63d}$$

$$(S_k + \sigma P_k, P_k) \text{ has no real eigenvalues less than zero.} \tag{4.63e}$$

Note that $N_k$ is symmetric positive definite. Thus, if $\lambda_{\min}^{(k)} > 0$ and $z_k = -S_k^{-1} h_k$ has $z_k^{\mathrm{T}} N_k z_k \leq \delta^2$, then $z_k$ satisfies the above conditions, and can be used to form the approximation $x_k$ to the trust-region problem. This corresponds to the solution of the trust-region problem where $\sigma = 0$. If this $z_k$ fails to satisfy the above conditions, then $\sigma_k > 0$ and $z_k^{\mathrm{T}} N_k z_k = \delta^2$. Let $z_k(\sigma) = -(S_k + \sigma P_k)^{-1} h_k$ and

$$\psi(\sigma) = \sqrt{z_k(\sigma)^{\mathrm{T}} N_k z_k(\sigma)} - \delta.$$

The following result shows that, under suitable conditions, the function $\psi$ has a zero on $(-\lambda_{\min}^{(k)}, \infty)$.

**Theorem 4.8.4.** *If $P_k$ is nonsingular, that is all the diagonals of $P_k$ are positive, and the system*

*$(S_k - \lambda_{\min}^{(k)} P_k) z_k = -h_k$ is not compatible, then $\psi(\sigma)$ has at least one root on the interval $(-\lambda_{\min}, \infty)$.*

*Proof.* Without loss of generality, assume the matrix $S_k$ is upper triangular, i.e., there are no complex eigenvalues, and that $N_k = I$. To simplify notation, drop the iteration index $k$. Assume the eigenvalue $\lambda_{\min}$ has algebraic multiplicity $m$. The generalized Schur decomposition can be constructed so that $\lambda_{\min}$ corresponds to the lower right $m \times m$ block. Then,

$$z(\sigma) = -(S + \sigma P)^{-1} h$$

can be solved via backward substitution, i.e.,

$$z_{k-j+1} = \frac{h_{k-j+1} - \sum_{i=k-j+2}^{n} (S_{k-j+1,i} + \sigma P_{k-j+1,i}) z_i}{S_{k-j+1,k-j+1} + \sigma T_{k-j+1,k-j+1}}$$

for $j = 1, \ldots, k$. By the incompatibility assumption, $h_{k-m+1}$ through $h_k$ are not equal to zero. So,

$$z_k = \frac{h_k}{S_{k,k} + \sigma P_{k,k}}$$

is a rational function of $\sigma$. Proceeding via induction, it is straightforward to show that

$$z_{k-j+1} = \frac{q_j(\sigma)}{p_j(\sigma)},$$

where $q_j$ is a polynomial of degree not exceeding $j-1$, and $p_j$ is a polynomial of degree $j$ with largest root $-\lambda_{\min}$. Then

$$z(\sigma)^{\mathrm{T}} z(\sigma) = \sum_{j=1}^{k} \frac{q_j(\sigma)^2}{p_j(\sigma)^2}.$$

Then $\lim_{\sigma \to -\lambda_{\min}} z(\sigma)^{\mathrm{T}} z(\sigma) = \infty$, as $\lim_{\sigma \to -\lambda_{\min}} z_k(\sigma)^2 = \infty$, and all other terms in the sum are nonnegative, and $\lim_{\sigma \to \infty} z(\sigma)^{\mathrm{T}} z(\sigma) = 0$. Now, $z(\sigma)^{\mathrm{T}} z(\sigma)$ is continuous and nonnegative, therefore $\sqrt{z(\sigma)^{\mathrm{T}} z(\sigma)}$ is continuous. By the intermediate value theorem, $\psi(\sigma)$ must then have at least one zero in $(-\lambda_{\min}, \infty)$. □

Unlike the standard trust problem, $\psi$ may in fact have multiple roots in $(-\lambda_{\min}, \infty)$. Fortunately, this does not seem to cause issues in practice and is rarely the case. Like the Moré-Sorensen algorithm, a safeguarded Newton's method can be performed on the function

$$\phi(\sigma) = \frac{1}{\delta} - \frac{1}{||z_k(\sigma)||_{N_k}}$$

to find such a root.

It is worth mentioning that the condition that $P_k$ be nonsingular will almost always be true in practice. If this fails to hold at some iteration $k$, solving the subproblem can be skipped. The updated solution can be taken to be $x_{k+1} = \alpha(x_k + v_{k+1})$, where $\alpha \geq 0$ is chosen so that the approximation does not exceed the trust-region boundary, and $\sigma_{k+1}$ is taken to be zero if $x_{k+1}$ is on the interior of the trust region, and $\sigma_{k+1} = -(g^{\mathrm{T}} x_{k+1} + x_{k+1}^{\mathrm{T}} H x_{k+1})/\delta^2$ if $x_{k+1}$ is on the boundary. A more difficult issue is the compatibility condition. This corresponds to the hard case in the standard trust-region problem. To illustrate the situation, assume that $\lambda_{\min}^{(k)}$ is a simple eigenvalue appearing in the upper left entry of $S_k$ and $P_k$, and that $(S_k - \lambda_{\min}^{(k)} P_k) z_k = -h_k$ is compatible. Then $h_k \in \mathrm{range}(S_k - \lambda_{\min}^{(k)} P_k) \perp \mathrm{null}(S_k^{\mathrm{T}} - \lambda_{\min}^{(k)} P_k^{\mathrm{T}})$. Let $v_{\min}^{(k)}$ be the left eigenvector of $(S_k - \lambda_{\min}^{(k)} P_k)$ corresponding to $\lambda_{\min}^{(k)}$. Then the system is compatible if and only if $h_k^{\mathrm{T}} v_{\min}^{(k)} = 0$. Note that the corresponding right eigenvector is $u_{\min}^{(k)} = e_1$. Let $s_k$ be the solution to the following problem:

$$\min_{s \in \mathbb{R}^k} ||s||_{N_k}$$
$$\text{subject to} \quad (S_k - \lambda_{\min}^{(k)} P_k) s = -h_k.$$

If $\lambda_{\min}^{(k)}$ is simple, then this corresponds to a one-dimensional minimization problem. If $||s_k||_{N_k} \leq \delta$, then $s_k + \tau u_{\min}^{(k)}$ satisfies (4.63) for some $\tau$. Otherwise, $\psi$ must have at least one root in $(-\lambda_{\min}^{(k)}, \infty)$, which can be solved for via Newton's method. Using the computed Schur decomposition, the left and right eigenvectors are straightforward to find. Therefore, the hard case can be checked upfront before running Newton's method. Furthermore, the Schur decomposition implies that $(S_k + \sigma P_k)$ will always be nearly upper triangular. Thus any systems of the form $(S_k + \sigma P_k) z_k = -h_k$ are trivial to solve. Therefore, the termination criteria used in this method are more straightforward than that of the Moré Sorensen method, i.e. once the $\sigma = 0$ and the hard case are checked for, the safeguarded Newton's method is applied to $\phi(\sigma)$ until $|\psi(\sigma)| \leq \varepsilon\delta$ and $||z(\sigma)||_{N_k} \leq (1 + \varepsilon)\delta$ for some tolerance $\varepsilon > 0$.

The scheme for updating the interval in which $\sigma$ belongs is equivalent to the Moré-Sorensen algorithm. However, the initial choice of interval $[\sigma_l, \sigma_u]$ is less obvious. As $\lambda_{\min}$ can be inferred from the Schur decomposition, $\sigma_l$ is set to $\max\{0, -\lambda_{\min}\}$. On the other hand, there is no simple expression with which to infer a choice of $\sigma_u$. Instead, an iterative approach is taken. Choosing $\sigma_0 = \max\{\tau, \sigma_l\}$ for some $\tau > 0$, and $\sigma_j = \gamma^j \sigma_0$ for some $\gamma > 0$, a sequence $\{\psi(\sigma_j)\}$ can be computed until a $j$ with $\psi(\sigma_j) < 0$ is found. Setting $\sigma_u = \sigma_j$ then creates a viable initial interval. The modified Moré-Sorensen algorithm can then proceed. The upper quasi-triangular structure of $S_k + \sigma P_k$ and low dimension imply that this initial iterative procedure only negligibly impacts the run time. Nevertheless, it is still recommended to choose $\gamma$

to be large enough that an acceptable $\sigma_u$ can be quickly found.

The final issue to be discussed is the choice of $\alpha_k$ and $\beta_k$. A simple option is $\alpha_k = 0$ and $\beta_k = 1$, which experimentally yield the most consistent results. However, other choices seem best suited for scalars in the interior of the spectrum of $(K, T)$, as would be the case for using a Jacobi-Davidson method to find interior eigenvalues.

This choice of $\alpha_k$ and $\beta_k$ yields a correction equation (4.62) that is not symmetric and, therefore, cannot be solved with preconditioned conjugate gradient. This, however, is in line with the primary motivation of this method. Recall that $K$ and $T$, in the case $i = 2$, form a saddle point system, and in the case $i = 1$, create a system equivalent to a saddle point system under column scaling. Unlike the locally-optimal preconditioned conjugate-gradient trust-region method, preconditioners no longer need to be limited to the form $\begin{pmatrix} G & J^{\mathrm{T}} \\ J & -D \end{pmatrix}$. Any preconditioner can be used, such as the incomplete LU decomposition or the incomplete signed Cholesky decomposition if $i = 2$. As both preconditioners were developed with GMRES in mind, restarted GMRES is used to solve the correction equation. However, other methods, such as CGS or BiCGSTAB, could be used instead of GMRES.

As was the case with the standard JDTR algorithm, it is beneficial to set a fixed number of initial iterations in which the correction equation is solved with $\sigma_i$ replaced with $\mu$ such that $H^M + \mu B^{(i)}$ is positive definite. This is equivalent to the requirement placed on the shift of the SIGLTR algorithm. Equation (4.30) suggests using

$$\mu = \frac{g^{\mathrm{T}} g}{\delta ||g||_B} - \lambda_{\mathrm{est}},$$

where $\lambda_{\mathrm{est}}$ is an estimate of the lowest eigenvalue of $(H^M, B^{(i)})$. Recall that $H^M + \sigma B^{(1)}$ is positive definite if and only if $H + \sigma I + \frac{1}{\sigma} J^{\mathrm{T}} D^{-1} J$ is positive definite, and $H^M + \sigma B^{(2)}$ is positive definite if and only if $H + \sigma I + (1 + \sigma) J^{\mathrm{T}} D^{-1} J$ is positive definite. This implies that if $H + \sigma I$ is positive definite, then $H^M + B^{(i)}$ for both values of $i$. Thus, finding an estimate of the lowest eigenvalue of $H$ provides a reliable choice of $\mu$. One simple method would be to use a few iterations of the Lanczos process to find such an estimate. Let $\lambda_{\mathrm{est}}(H)$ be the approximate eigenvalue found by a few iterations of the Lanczos process. Then $\lambda_n(H) \leq \lambda_{\mathrm{est}}(H)$, and $\mu = \frac{g^{\mathrm{T}} g}{\delta ||g||_B} - \lambda_{\mathrm{est}}(H)$ is not guaranteed to yield a positive definite matrix. Let $\ell$ denote the lower bound on $\lambda_n(H)$ found by the Gerschgorin circle theorem. Then $\ell \leq \lambda_n(H)$. Most likely, $\ell$ will be a significant underestimate of $\lambda_n(H)$. This motivates the choice of

$$\frac{g^{\mathrm{T}} g}{\delta ||g||_B} - ((1 - \tau)\lambda_{\mathrm{est}}(H) + \tau \ell) \tag{4.64}$$

for some small positive value of $\tau$. If this choice of $\mu$ fails to yield a positive definite matrix, the choice

$$\frac{g^{\mathrm{T}}g}{\delta||g||_B} - \ell \tag{4.65}$$

is guaranteed to succeed. These estimates work well for both JDQZTR and SIGLTR applied to doubly-augmented trust-region problems.

The Jacobi-Davidson QZ trust-region algorithm is best suited for doubly-augmented trust-region problems in which the dimension is so large that even one factorization of a matrix of the form $\begin{pmatrix} G & J^{\mathrm{T}} \\ J & -D \end{pmatrix}$ is infeasible. If such a factorization can be performed efficiently, then the other methods discussed thus far tend to outperform this method. No problems in the CUTEST NLP collection required using this method in experiments. However, some artificially constructed doubly-augmented trust-region problems could rapidly converge using this method. In contrast, the other methods could not be applied, as the single matrix factorization exhausted all available memory before completing. This method also suffers all of the issues from ill-conditioning that GMRES suffers from. Namely, if the correction equation using a given preconditioner is ill-conditioned, convergence can be quite slow when far from the solution. In these situations, SIGLTR and locally optimal preconditioned conjugate-gradient tended to perform comparably to this method, even if the initial factorization took a large percentage of the execution time. The key takeaway is that the collection of methods presented here should provide a suitable means of solving all but the most extreme of doubly-augmented trust-region subproblems.

# Chapter 5

# The All-Shifted Primal-Dual Penalty-Barrier Trust-Region Method

## 5.1 Introduction

Consider the inequality-constrained optimization problem

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ &\text{subject to} \quad c(x) \geq 0, \end{aligned} \tag{5.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are assumed to be twice continuously differentiable, and the notation $v \geq 0$ is defined to act element-wise on a vector $v$. This problem can be reformulated to only include simple bounds by introducing a vector $s \in \mathbb{R}^m$ of *slack variables*. The resulting problem is given by

$$\begin{aligned} &\min_{x \in \mathbb{R}^n, s \in \mathbb{R}^m} f(x) \\ &\text{subject to} \quad c(x) - s = 0, \\ &\text{and} \quad s \geq 0. \end{aligned} \tag{5.2}$$

Note that any additional equality constraints and bounds on $x$ can easily be incorporated into this formulation. Most practical implementations of constrained optimization algorithms explicitly work with problems in the generic form

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ &\text{subject to} \quad \ell_x \leq x \leq u_x, \\ &\text{and} \quad \ell_s \leq c(x) \leq u_s, \end{aligned}$$

or the slack formulation

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ \text{subject to} \quad &\ell_x \leq x \leq u_x, \\ &\ell_s \leq s \leq u_s, \\ \text{and} \quad &c(x) - s = 0, \end{aligned} \tag{5.3}$$

where $-\infty \leq \ell_x \leq u_x \leq \infty$ and $-\infty \leq \ell_s \leq u_s \leq \infty$, however the problem formats (5.1) and (5.2) are sufficient for any theoretical analysis.

## 5.2 Shifted Primal-Dual Interior Point Algorithm

Consider the inequality and equality constrained problem (5.2). A vector $v^* = (x^*, s^*, y^*, w^*)$ is a first-order KKT point of (5.2) if

$$c(x^*) - s^* = 0, \quad s^* \geq 0, \tag{5.4a}$$

$$g(x^*) - J(x^*)^{\mathrm{T}} y^* = 0, \quad y^* - w^* = 0 \tag{5.4b}$$

$$s^* \cdot w^* = 0, \quad w^* \geq 0. \tag{5.4c}$$

The vectors $y^* \in \mathbb{R}^m$ and $w^* \in \mathbb{R}^m$ are the Lagrange multipliers for the equality and inequality constraints, respectively. Let $v_k = (x_k, s_k, y_k, w_k)$ denote the $k$-th primal-dual iterate computed by the algorithm, with the goal that the limit points of $\{v_k\}_{k=0}^{\infty}$ are first-order KKT points that satisfy (5.4).

In a typical interior-point algorithm, the complementarity condition (5.4c) is perturbed to be

$$s \cdot w = \mu^B e,$$

where $e$ is the vector of all ones and $\mu^B > 0$ is the *barrier parameter*. This has the effect of forcing the variables $s_k$ and $w_k$ to remain in the set $\{s > 0, w > 0\}$, i.e., the strictly feasible set. In [13], Gill et al. propose perturbing the complementarity condition instead by

$$s \cdot w = \mu^B (w^E - w),$$

where $w^E$ is an estimate of the optimal values of $w$. This has the effect of shifting the primal constraint

$s \geq 0$ to $s \geq -\mu^B e$. In [17], Gill et al. further suggest perturbing the complementarity condition by

$$s \cdot w = \mu^B(s^E - s) + \mu^B(w^E - w),$$

where $s^E$ is an estimate of the optimal value of $s$. This can be rewritten as

$$(s + \mu^B e) \cdot (w + \mu^B e) = \mu^B s^E + \mu^B w^E + (\mu^B)^2 e.$$

Perturbing the complementarity condition in such a manner, therefore, has the effect of shifting the boundary of the strictly feasible set to $\{s + \mu^B e = 0, w + \mu^B = 0\}$, allowing iterates to temporarily leave the feasible set on the way to the optimal value. Moreover, as the optimal values of $s$ and $w$ will, in many cases, lie on the boundary of the feasible set, this formulation avoids requiring $\mu^B$ to be driven to zero.

The equality constraints are similarly perturbed using a primal-dual augmented Lagrangian approach, as in [15]. The perturbed first-order KKT conditions are then

$$\nabla f(x) - J(x)^{\mathrm{T}} y = 0, \qquad y - w = 0 \tag{5.5a}$$

$$c(x) - s = \mu^P(y^E - y), \qquad s \geq 0 \tag{5.5b}$$

$$s \cdot w = \mu^B(w^E - w) + \mu^B(s^E - s), \qquad w \geq 0, \tag{5.5c}$$

where $\mu^P > 0$ is the *penalty parameter*, and $y^E$ is an estimate of the optimal value of $y$. In a neighborhood of a first-order KKT point, it is well known that computing a search direction as the solution of Newton's equations for a zero of the perturbed optimality conditions provides the ideal local convergence rates commonly associated with Newton's method. At the same time, to ensure convergence from an arbitrary starting point, any algorithm must include a strategy for deciding whether one iterate is preferable to

another. To that end, the following merit function is introduced.

$$
\begin{aligned}
M(x,s,y,w;s^E,y^E,w^E,\mu^P,\mu^B) = {} & \underbrace{f(x)}_{A}\underbrace{-(c(x)-s)^{\mathrm{T}}y^E}_{B} \\
& \underbrace{+\frac{1}{2\mu^P}||c(x)-s||^2}_{C}+\underbrace{\frac{1}{2\mu^P}||c(x)-s+\mu^P(y-y^E)||^2}_{D} \\
& \underbrace{-2\sum_{i=1}^{m}(\mu^B w_i^E+\mu^B s_i^E+(\mu^B)^2)\ln(s_i+\mu^B)}_{E}+\underbrace{2\mu^B s^{\mathrm{T}}e}_{F} \\
& \underbrace{-\sum_{i=1}^{m}(\mu^B w_i^E+\mu^B s_i^E+(\mu^B)^2)\ln(w_i+\mu^B)}_{G}+\underbrace{\mu^B w^{\mathrm{T}}e}_{H}+\underbrace{w^{\mathrm{T}}s}_{I}. \qquad (5.6)
\end{aligned}
$$

It will be shown that, in a neighborhood of a minimizer of (5.2) satisfying certain second-order optimality conditions, Newton's equations for a zero of the perturbed optimality conditions (5.5) are equivalent to the Newton equations for a minimizer of $M$. Additionally, it will be shown that if the parameters $s^E$, $y^E$, $w^E$, $\mu^P$ and $\mu^B$ are updated appropriately, then stationary points of $M$ have convenient properties that can be used to construct a globally convergent method to solve (5.2).

Before proceeding, some additional notation is needed. Let $S$ and $W$ denote the diagonal matrices $\mathrm{diag}(s)$ and $\mathrm{diag}(w)$, respectively. Let

$$
\pi^y = y^E - \frac{1}{\mu^P}\big(c(x)-s\big), \quad \text{and} \quad \pi^w = (S+\mu^B I)^{-1}\left(\mu^B w^E - \mu^B(s-s^E)\right), \qquad (5.7)
$$

and

$$
D_y = \mu^P I, \quad \text{and} \quad D_w = \big(S+\mu^B I\big)\big(W+\mu^B I\big)^{-1}. \qquad (5.8)
$$

Note that $y=\pi^y$ and $w=\pi^w$ are the stationary points of $M$ with the variables $x$ and $s$ fixed. Then $\nabla M$ may be written as

$$
\nabla M = \begin{pmatrix} g(x)-J(x)^{\mathrm{T}}(2\pi^y-y) \\ (2\pi^y-y)-(2\pi^w-w) \\ D_y(y-\pi^y) \\ D_w(w-\pi^w) \end{pmatrix}. \qquad (5.9)
$$

Let $W_{B} = W + \mu^{B} I$ and $\Pi_{B}^{w} = \operatorname{diag}(\pi^{w} + \mu^{B} e)$. The penalty barrier function Hessian can be written as

$$\nabla^2 M = \begin{pmatrix} H + 2J(x)^{\mathrm{T}} D_y^{-1} J(x) & -2J(x)^{\mathrm{T}} D_y^{-1} & J(x)^{\mathrm{T}} & 0 \\ -2D_y^{-1} J(x) & 2(D_y^{-1} + D_w^{-1} W_{B}^{-1} \Pi_{B}^{w}) & -I & I \\ J & -I & D_y & 0 \\ 0 & I & 0 & D_w W_{B}^{-1} \Pi_{B}^{w} \end{pmatrix}, \tag{5.10}$$

where $H = H(x, 2\pi^{y} - y)$ is the Hessian of the Lagrangian with respect to the $x$ variables, i.e. $H(x, y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 c_i(x)$.

### 5.2.1 Minimizing the Primal-Dual Merit Function

In [13] and [17], Gill et al. propose a method for minimizing the merit function $M$ using a suitable line search approach. However, the method presented here differs in that a trust-region approach is used instead of a line search procedure. With the trust-region algorithms introduced in Chapter 4, a pure trust-region approach has become significantly more feasible for use in large-scale optimization, and unlike in [12], a solution to the trust-region subproblem can be followed up with solving a trust-region subproblem with a reduction of the trust-region radius, as opposed to a line search.

In what follows, consider the parameters $s^{E}$, $y^{E}$, $w^{E}$, $\mu^{P}$, and $\mu^{B}$ to be fixed. Let

$$q_k(v) = M(v_k) + \nabla M(v_k)^{\mathrm{T}}(v - v_k) + \frac{1}{2}(v - v_k)^{\mathrm{T}} H_{k}^{M}(v - v_k)$$

be a local quadratic model of the merit function $M$, and let $Q_k(d_v) = q_k(v_k + d_v) - q_k(v_k)$. At each iteration, the following trust-region subproblem is solved:

$$\begin{aligned} &\min_{d_v \in \mathbb{R}^{n+3m}} Q_k(d_v) = d_v^{\mathrm{T}} \nabla M(v_k) + \frac{1}{2} d_v^{\mathrm{T}} H_{k}^{M} d_v \\ &\text{subject to} \quad d_v^{\mathrm{T}} B_k d_v \leq \delta^2, \end{aligned} \tag{5.11}$$

where $B_k$ is a positive definite matrix, and

$$H^{M} = \begin{pmatrix} H(x,y) + 2J(x)^{\mathrm{T}} D_y^{-1} J(x) & -2J(x)^{\mathrm{T}} D_y^{-1} & J(x)^{\mathrm{T}} & 0 \\ -2D_y^{-1} J(x) & 2(D_y^{-1} + D_w^{-1}) & -I & I \\ J & -I & D_y & 0 \\ 0 & I & 0 & D_w \end{pmatrix} \tag{5.12}$$

is the matrix obtained by replace $\pi^y$ and $\pi^w$ with $y$ and $w$, respectively. Note that $H^M$ has a doubly-augmented structure. Before defining $B_k$, suppose that the problem being solved has $H(x, y)$ strongly positive definite and that Newton's method can be directly applied to $M$. Then the search direction at each step satisfies

$$H_k^M d_v = -\nabla M_k.$$

Following Section 4.8, this can be transformed to

$$\begin{pmatrix} H(x,y) & 0 & -J(x) & 0 \\ 0 & 0 & I & -I \\ -J(x) & I & D_y & 0 \\ 0 & -I & 0 & D_w \end{pmatrix} \begin{pmatrix} d_x \\ d_s \\ d_y \\ d_w \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ y - w \\ D_y(\pi^y - y) \\ D_w(\pi^w - w) \end{pmatrix}. \tag{5.13}$$

This system should not be solved directly. Instead, the system can be further reduced to

$$\begin{pmatrix} H(x,y) & -J(x) \\ -J(x) & -(D_y + D_w) \end{pmatrix} \begin{pmatrix} d_x \\ d_y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ D_y(\pi^y - y) + D_w(\pi_w - w) \end{pmatrix} \tag{5.14}$$

and

$$d_w = y - w + d_y, \quad \text{and} \quad d_s = \mu^B W_B^{-1}(w^E + s^E - s) - D_w(y + d_y).$$

It can be shown that the matrix $H^M$ is positive definite if and only if the matrix appearing in (5.14) has exactly $n$ positive and $m$ negative eigenvalues. Depending on the definition of $B_k$, a similar procedure can be followed to simplify systems of the form

$$(H_k^M + \sigma B_k) d_v = \nabla M_k.$$

The choice of model function Hessian $H^M$ can be further justified by examining the *path-following equations*. Consider the path-following equations of the perturbed KKT conditions (5.5)

$$F(x, s, y, w; s^E, y^E, w^E, \mu^P, \mu^B) = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ y - w \\ c(x) - s + \mu^P(y - y^E) \\ s \cdot w + \mu^B(s - s^E) + \mu^B(w - w^E) \end{pmatrix} = 0. \tag{5.15}$$

A zero of $F$ satisfying $s > -\mu^B e$ and $w > -\mu^B e$ approximates a solution of problem (5.2). The approximation becomes increasingly accurate as $\mu^P(y - y^E) \to 0$, $\mu^B(s - s^E) \to 0$, and $\mu^B(w - w^E) \to 0$. The Newton equations for $F$ are

$$\begin{pmatrix} H(x,y) & 0 & -J(x) & 0 \\ 0 & 0 & I & -I \\ J(x) & -I & D_y & 0 \\ 0 & W + \mu^B I & 0 & S + \mu^B I \end{pmatrix} \begin{pmatrix} d_x \\ d_s \\ d_y \\ d_w \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ y - w \\ c(x) - s + \mu^P(y - y^E) \\ s \cdot w + \mu^B(s - s^E) + \mu^B(w - w^E) \end{pmatrix}.$$

A simple row scaling shows that these equations are equivalent to (5.13).

In [12], Gertz et al. propose a primal dual interior point trust-region algorithm in which the trust-region matrix $B_k$ is defined as

$$B_k^{(1)} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & D_y & 0 \\ 0 & 0 & 0 & D_w \end{pmatrix}. \tag{5.16}$$

While this matrix is straightforward to work with, the matrix pencils $(H_k^M, B_k^{(1)})$ are poorly conditioned because of the doubly-augmented structure of $H_k^M$. The matrix $B_k^{(1)}$ is referred to as the *diagonal trust-region matrix*. Consider the alternative trust-region matrix

$$B_k^{(2)} = \begin{pmatrix} I + 2J(x)^{\mathrm{T}} D_y^{-1} J(x) & -2J(x)^{\mathrm{T}} D_y^{-1} & J(x)^{\mathrm{T}} & 0 \\ -2D_y^{-1} J(x) & I + 2(D_y^{-1} + D_w^{-1}) & -I & I \\ J & -I & D_y & 0 \\ 0 & I & 0 & D_w \end{pmatrix}, \tag{5.17}$$

which is called the *doubly-augmented trust-region matrix*. Both matrices are positive definite, and systems of the form $(H_k^M + \sigma B_k^{(i)})d_v = -\nabla M_k$ can be reduced similarly to (5.14) for both $i = 1$ and 2.

First, consider the case $i = 1$, so for any $\mu$,

$$H^M + \mu B^{(1)} = \begin{pmatrix} H(x,y) + \mu I + 2J(x)^{\mathrm{T}} D_y^{-1} J(x) & -2J(x)^{\mathrm{T}} D_y^{-1} & J(x)^{\mathrm{T}} & 0 \\ -2D_y^{-1} J(x) & \mu I + 2(D_y^{-1} + D_w^{-1}) & -I & I \\ J & -I & (1+\mu I) D_y & 0 \\ 0 & I & 0 & (1+\mu I) D_w \end{pmatrix}.$$

Regardless of which trust-region method is used, equations of the form $(H^M + \mu B^{(1)})v = r$ should not be solved directly. Let $r = (r_x, r_s, r_y, r_w)$, and $v = (x, s, y, w)$. Let $J = J(x)$, $H = H(x,y)$, and

$$R = \begin{pmatrix} I & 0 & 2J^{\mathrm{T}} D_y^{-1} & 0 \\ 0 & I & -2D_y^{-1} & 2D_w^{-1} \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & -I \end{pmatrix}.$$

Then

$$R(H^M + \mu B^{(1)})v = \begin{pmatrix} H + \mu I & 0 & -(1+2\mu)J^{\mathrm{T}} & 0 \\ 0 & \mu I & (1+2\mu)I & -(1+2\mu)I \\ -J & I & -(1+\mu)D_y & 0 \\ 0 & -I & 0 & -(1+\mu)D_w \end{pmatrix} \begin{pmatrix} x \\ s \\ y \\ w \end{pmatrix} = \begin{pmatrix} r_x + 2J^{\mathrm{T}} D_y^{-1} r_y \\ r_s - 2D_y^{-1} r_y + 2D_w^{-1} r_w \\ -r_y \\ -r_w \end{pmatrix}.$$

Let $\widehat{\mu} = \frac{1+\mu}{1+2\mu}$. Then the above equation can be symmetrized to become

$$\begin{pmatrix} H + \mu I & 0 & -J^{\mathrm{T}} & 0 \\ 0 & \mu I & I & -I \\ -J & I & -\widehat{\mu} D_y & 0 \\ 0 & -I & 0 & -\widehat{\mu} D_w \end{pmatrix} \begin{pmatrix} x \\ s \\ (1+2\mu)y \\ (1+2\mu)w \end{pmatrix} = \begin{pmatrix} r_x + 2J^{\mathrm{T}} D_y^{-1} r_y \\ r_s - 2D_y^{-1} r_y + 2D_w^{-1} r_w \\ -r_y \\ -r_w \end{pmatrix}.$$

A row and column permutation then gives

$$\begin{pmatrix} H + \mu I & -J^{\mathrm{T}} & 0 & 0 \\ -J & -\widehat{\mu} D_y & I & 0 \\ 0 & I & \mu I & -I \\ 0 & 0 & -I & -\widehat{\mu} D_w \end{pmatrix} \begin{pmatrix} x \\ (1+2\mu)y \\ s \\ (1+2\mu)w \end{pmatrix} = \begin{pmatrix} r_x + 2J^{\mathrm{T}} D_y^{-1} r_y \\ -r_y \\ r_s - 2D_y^{-1} r_y + 2D_w^{-1} r_w \\ -r_w \end{pmatrix}.$$

These equations, which are almost block-diagonal with the majority of nonzero blocks diagonal, can be further simplified. Let $\widehat{D} = (\mu I + \frac{1}{\widehat{\mu}} D_w^{-1})^{-1}$, and

$$\widehat{R} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & -\widehat{D} & \frac{1}{\widehat{\mu}} D_w^{-1} \widehat{D} \\ 0 & 0 & I & -\frac{1}{\widehat{\mu}} D_w^{-1} \\ 0 & 0 & 0 & I \end{pmatrix}.$$

Left multiplying by $\widehat{R}$ gives

$$\begin{pmatrix} H + \mu I & -J^{\mathrm{T}} & 0 & 0 \\ -J & -\widehat{\mu} D_y - \widehat{D} & 0 & 0 \\ 0 & I & \widehat{D}^{-1} & 0 \\ 0 & 0 & -I & -\widehat{\mu} D_w \end{pmatrix} \begin{pmatrix} x \\ (1+2\mu)y \\ s \\ (1+2\mu)w \end{pmatrix}$$
$$= \begin{pmatrix} r_x + 2J^{\mathrm{T}} D_y^{-1} r_y \\ -r_y - \widehat{D}(r_s - 2D_y^{-1} r_y + 2D_w^{-1} r_w) - \frac{1}{\widehat{\mu}} D_w^{-1} \widehat{D} r_w \\ r_s - 2D_y^{-1} r_y + 2D_w^{-1} r_w + \widehat{\mu} D_w r_w \\ -r_w \end{pmatrix}.$$

The upper left block can be used to solve for $x$ and $y$, and then substitution can be used to find the remaining variables. Thus, the $(n + 3m) \times (n + 3m)$ dimensional system can be converted into an $(n + m) \times (n + m)$ dimensional saddle-point system.

Next, consider the case $i = 2$, so for any $\mu$,

$$(H^M + \mu B^{(2)}) = (1 + \mu) \begin{pmatrix} \frac{1}{1+\mu}(H + \mu I) + 2J^{\mathrm{T}} D_y^{-1} J & -2J^{\mathrm{T}} D_y^{-1} & J^{\mathrm{T}} & 0 \\ -2D_y^{-1} J & \frac{\mu}{1+\mu} I + 2(D_y^{-1} + D_w^{-1}) & -I & I \\ J & -I & D_y & 0 \\ 0 & I & 0 & D_w \end{pmatrix}.$$

Let $R$ be defined the same as before, so that

$$R(H^M + \mu B^{(2)})v = (1+\mu)\begin{pmatrix} \frac{1}{1+\mu}(H+\mu I) & 0 & -J^{\mathrm{T}} & 0 \\ 0 & \frac{\mu}{1+\mu}I & I & -I \\ -J & I & -D_y & 0 \\ 0 & -I & 0 & -D_w \end{pmatrix}\begin{pmatrix} x \\ s \\ y \\ w \end{pmatrix} = \begin{pmatrix} r_x + 2J^{\mathrm{T}}D_y^{-1}r_y \\ r_s - 2D_y^{-1}r_y + 2D_w^{-1}r_w \\ -r_y \\ -r_w \end{pmatrix}.$$

A row and column permutation then gives

$$(1+\mu)\begin{pmatrix} \frac{1}{1+\mu}(H+\mu I) & -J^{\mathrm{T}} & 0 & 0 \\ -J & -D_y & I & 0 \\ 0 & I & \frac{\mu}{1+\mu}I & -I \\ 0 & 0 & -I & -D_w \end{pmatrix}\begin{pmatrix} x \\ y \\ s \\ w \end{pmatrix} = \begin{pmatrix} r_x + 2J^{\mathrm{T}}D_y^{-1}r_y \\ -r_y \\ r_s - 2D_y^{-1}r_y + 2D_w^{-1}r_w \\ -r_w \end{pmatrix}.$$

Let $\widehat{D} = (\frac{\mu}{1+\mu}I + D_w^{-1})^{-1}$, and

$$\widehat{R} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & -\widehat{D} & D_w^{-1}\widehat{D} \\ 0 & 0 & I & -D_w^{-1} \\ 0 & 0 & 0 & I \end{pmatrix}.$$

Left multiplying by $\widehat{R}$ gives

$$(1+\mu)\begin{pmatrix} \frac{1}{1+\mu}(H+\mu I) & -J^{\mathrm{T}} & 0 & 0 \\ -J & -D_y - \widehat{D} & 0 & 0 \\ 0 & I & \widehat{D}^{-1} & 0 \\ 0 & 0 & -I & -D_w \end{pmatrix}\begin{pmatrix} x \\ y \\ s \\ w \end{pmatrix}$$

$$= \begin{pmatrix} r_x + 2J^{\mathrm{T}}D_y^{-1}r_y \\ -r_y - \widehat{D}(r_s - 2D_y^{-1}r_y + 2D_w^{-1}r_w) - D_w^{-1}\widehat{D}r_w \\ r_s - 2D_y^{-1}r_y + 2D_w^{-1}r_w + D_w r_w \\ -r_w \end{pmatrix}.$$

The upper left block can be used to solve for $x$ and $y$, and then substitution can be used to find the remaining variables. Thus, the $(n+3m)\times(n+3m)$ dimensional system can be converted into an $(n+m)\times(n+m)$ dimensional saddle-point system.

As discussed in Section 2.5, trust-region methods for unconstrained optimization problems have excellent theoretical properties and work quite well in practice, often exhibiting convergence in fewer iterations than inertia-controlling line-search strategies at the cost of the larger computational overhead required to solve the trust-region subproblems. However, some difficulties emerge when applying these results to the minimization of $M$. First, because of the barrier terms in the merit function, $M$ is not defined outside the strict interior of the shifted feasible set. The merit function $M$ can be defined as infinity off this set. Doing so ensures that the trust-region method will reject any steps outside this set. Because of the improvements in trust-region algorithms presented in Chapter 4, it is possible to solve a subsequent trust-region subproblem with a reduced radius quickly. The trust-region step $v_{k+1} = v_k + d_v$ is accepted if the condition

$$M(v_k) - M(v_{k+1}) \geq -\eta_1 Q_k(d_v) \tag{5.18}$$

holds, where $\eta_1 \in (0, \frac{1}{2})$. Due to the shifted constraints, iterates $v_{k+1}$ can satisfy this condition even if they lie outside the feasible set. This includes the boundary of the feasible set, which is excluded in typical interior methods. As usual, the restriction of $\eta_1$ to $(0, \frac{1}{2})$ prevents the sufficient decrease condition from interfering with the asymptotic convergence rate of the algorithm.

The trust-region algorithm used is presented in Algorithm 5.1

**Algorithm 5.1.** Merit Function Trust-Region Algorithm

Given constants $\eta_1$, $\eta_2$, $\gamma_C$, $\gamma_E$, $\delta_0$ such that $0 < \eta_1 < \eta_2 < 1$, $\eta_1 < 1/2$, $0 < \gamma_C < 1 < \gamma_E$, and $\delta_0 > 0$
$k \leftarrow 0$
**while** not converged **do**
    $d_k = \text{argmin}_{p \in \mathbb{R}^n} \{Q_k(d_v) \ : \ ||d_v||_{B_k^{(i)}} \leq \delta_k\}$
    $\rho_k = (M(v_k) - M(v_k + d_k))/Q_k(d_k)$
    **if** $\rho_k \geq \eta_1$ **then**
        $\widehat{v}_{k+1} = v_k + d_k$
        **if** $\rho_k \geq \eta_2$ **then**
            $\delta_{k+1} \leftarrow \max\{\delta_k, \gamma_E ||d_k||_{B_k^{(i)}}\}$
        **else**
            $\delta_{k+1} \leftarrow \delta_k$
        **end if**
        $s_{k+1} \leftarrow \max\{\widehat{s}_{k+1}, c(x_{k+1}) - \mu^P(y^E + \frac{1}{2}(w_{k+1} - y_{k+1}) + \mu^B e)$ ▷ Slack Reset
        $v_{k+1} \leftarrow (\widehat{x}_{k+1}, s_{k+1}, \widehat{y}_{k+1}, \widehat{w}_{k+1})$
    **else**
        $v_{k+1} \leftarrow v_k$
        $\delta_{k+1} \leftarrow \gamma_C ||d_k||_{B_k^{(i)}}$
    **end if**
    $k \leftarrow k + 1$
**end while**

An approximate solution $d_v$ to (5.11) is required to satisfy the following conditions:

$$Q_k(d_v) \leq \tau \min\{b_{k,1}, b_{k,2}\}, \tag{5.19a}$$

$$||d_v||_{B_k^{(i)}} \leq \delta_k, \text{ and} \tag{5.19b}$$

$$\text{either } \nabla M(v_k)^{\mathrm{T}} d_v < 0, \text{ or } \nabla M(v_k)^{\mathrm{T}} d_v \leq 0 \text{ and } d_v^{\mathrm{T}} H_k^M d_v < 0, \tag{5.19c}$$

where

$$b_{k,1} = -||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} \min\left\{\delta_k, \frac{||\nabla M(v_k)||_{(B_k^{(i)})^{-1}}}{||H_k^M||_{B_k^{(i)}}}\right\}, \tag{5.20}$$

and

$$b_{k,2} = \frac{1}{2}\delta_k^2 \min\{0, \lambda_{\min}(H_k^M, B_k^{(i)})\} \tag{5.21}$$

for some $\tau \in (0, 1)$. These bounds are the model decrease of $Q$ subject to the trust-region constraint along the steepest-descent direction induced by the trust-region norm and the leftmost eigenvector of $(H_k^M, B_k^{(i)})$, respectively. Compare this with Assumptions 2.5.7 and 2.5.9. In practice, the approximate solution to each trust-region subproblem will be very close to the true solution. However, this requirement is sufficient to achieve convergence. Line 13 of Algorithm 5.1 constitute an additional step known as a *slack reset*. This step is crucial to the convergence of the outer algorithm. It will be shown that the slack reset does not hinder the convergence of Algorithm 5.1 to a local minimizer of $M$.

### 5.2.2 Convergence Analysis

The following assumptions are made for the convergence analysis:

**Assumption 5.2.1.** *The functions $f$ and $c$ are twice continuously differentiable.*

**Assumption 5.2.2.** *The sequence of iterates $\{x_k\}$ is contained in a bounded set.*

Note the similarities between Assumptions 5.2.1 and 5.2.2 and Assumption 2.5.1 and 2.5.10. It is important to note that not every iteration of Algorithm 5.1 is successful. Recalling that $M$ is defined to be infinite outside the shifted feasible set, define the sets

$$\mathcal{S} = \{k : \rho_k \geq \eta_1\} \quad \text{and} \quad \mathcal{V} = \quad \{k : \rho_k \geq \eta_2\}$$

to be the set of successful and very successful trust-region iterations, respectively. Note that if $k \notin \mathcal{S}$, then $\delta_{k+1} \leq \delta_k$. The following result shows that $M$ satisfies the decrease condition $M(v_{k+1}) \leq M(v_k)$.

**Lemma 5.2.1.** *For each iteration $k \geq 0$, it holds that $M(v_{k+1}) \leq M(v_k)$.*

*Proof.* Before the slack reset, the sufficient decrease condition ensures that $M$ is decreasing. Thus, the only way the result can be false is if the slack reset can increase $M$. The vector $c(x_{k+1}) - \mu^P(y^E + \frac{1}{2}(w_{k+1} - y_{k+1}) + \mu^B e)$ is the unique minimizer of the sum of terms (B), (C), (D), (F), (H), and (I) in the definition of $M$, so the slack reset does not increase these terms. (A) is independent of $s$. The slack reset can only increase the individual entries of $s$. Thus the remaining terms can only decrease. Thus, the slack reset can only reduce the value of $M$. □

**Corollary 5.2.1.1.** *If $k \in \mathcal{S}$, then*

$$M(v_k) - M(v_{k+1}) \geq M(v_k) - M(\widehat{v}_{k+1}) \geq -\eta_1 Q(d_{v,k}).$$

The above inequality will allow the proofs of the following results to closely follow the results presented in Section 2.5. The following result demonstrates that most of the assumptions of section 2.5 are satisfied by the function $M$ and the model functions $q_k$.

**Lemma 5.2.2.** *The sequence of iterates $\{v_k\} = \{(x_k, s_k, y_k, w_k)\}$ computed by Algorithm 5.1 satisfies the following properties.*

1. *The sequences $\{s_k\}$, $\{c(x_k) - s_k\}$, $\{y_k\}$ and $\{w_k\}$ are bounded.*
2. *For each index $i \in \{1, \ldots, m\}$, it holds that*

$$\liminf_{k \geq 0} (s_k + \mu^B e)_i \geq 0 \quad \textit{and} \quad \liminf_{k \geq 0} (w_k + \mu^B)_i \geq 0.$$

3. *The sequences $\{\pi^y(x_k, s_k)\}$, $\{\pi^w(s_k)\}$, and $\nabla M(v_k)$ are bounded.*
4. *There exists a scalar $\bar{M}$ such that $M(v_k) \geq \bar{M}$ for all $K$.*
5. *The eigenvalues of the matrices $\{B_k^{(i)}\}$ are all positive, bounded away from zero, and bounded above. Therefore, there exists a positive constant $\kappa_1$ such that $(1/\kappa_1)||v|| \leq ||v||_{B_k^i} \leq \kappa_1 ||v||$ for all $v \in \mathbb{R}^{n+3m}$.*
6. *The sequence $\{||H_k^M||\}$ is bounded.*
7. *The sequence $\{||\nabla^2 M(v_k)||\}$ is bounded.*

*Proof.* Assume that the sequence $\{s_k\}$ is unbounded to establish a contradiction. By construction, $s_k + \mu^B e > 0$ for all $k$. Therefore, if $\{s_k\}$ is unbounded, then there exists a subsequence $\mathcal{K}$ and an index $i$ such that

$$\lim_{k \in \mathcal{K}} [s_k]_i = \infty \quad \text{and} \quad [s_k]_i \geq [s_k]_j \ \text{ for all } k \in \mathcal{K} \text{ and } j. \tag{5.22}$$

It will be shown that if this is the case, then $M$ goes to infinity, contradicting Lemma 5.2.1. It follows from (5.22), Assumption 5.2.2, and the continuity of $c$ that the term (A) in the definition of $M$ is bounded below for all $k$, the term (B) cannot go to $-\infty$ faster than $||s_k||$ on $\mathcal{K}$, and the term (C) goes to $\infty$ on $\mathcal{K}$ at the same rate as $||s_k||^2$. Furthermore, (D) is bounded below by zero. On the other hand, the barrier term (E) goes to $-\infty$ on $\mathcal{S}$ like $-\ln([s_k]_i + \mu^B)$. The terms (F), (H), and (I) can be rewritten as

$$(F) + (H) + (I) = (w_k + \mu^B e)^{\mathrm{T}}(s_k + \mu^B e) + \mu^B (s_k + \mu^B e)^{\mathrm{T}} e - 2(\mu^B)^2 e.$$

As $s_k + \mu^B e > 0$ and $w + \mu^B e > 0$ for all $k$ by construction, the sum $(F) + (H) + (I)$ is bounded below. Therefore, if (G) is bounded below, $M$ goes to $\infty$ on $M$, establishing the contradiction with lemma (5.2.1). If instead (G) goes to $-\infty$, then it follows that there exists an index $j$ of $w$ such that $[w_k]_j \to \infty$. In that case, the (H) term will go to $\infty$ faster than (G) goes to $-\infty$, so $M$ goes to $\infty$. Thus, $\{s_k\}$ must be bounded. Now, because $c$ is continuous and $\{x_k\}$ is bounded, $\{c(x_k) - s_k\}$ is bounded as well.

Next, it will be shown that $\{y_k\}$ is bounded. To establish a contradiction, assume that it is not, i.e., that there exists a subsequence $\mathcal{K}$ and index $i$ such that

$$\lim_{k \in \mathcal{K}} |[y_k]_i| = \infty \quad \text{and} \quad |[y_k]_i| \geq |[y_k]_j| \ \text{ for all } k \in \mathcal{K} \text{ and } j. \tag{5.23}$$

Terms (A), (B), and (C) are bounded below for all $k$, and (D) converges to $\infty$ at the same rate as $([y_k]_i)^2$. By the boundedness of $\{s_k\}$, (E) is bounded below. Similar to before, if (G) is bounded below, then $M$ goes to $\infty$, establishing the contradiction, and if not, i.e., if (G) goes to $-\infty$, it does so slower than (H) goes to $\infty$, so $M$ goes to $\infty$. Thus, $\{y_k\}$ is bounded.

Finally, $\{w_k\}$ is bounded, otherwise (G) may be unbounded below on a subsequence $\mathcal{K}$, and (H) goes to $\infty$ on $\mathcal{K}$ faster than (G) goes to $-\infty$.

Part 2 is similarly proved by contradiction, in that if it is false, $M$ goes to $\infty$. Suppose that $\liminf_{k \geq 0}(s_k + \mu^B e)_i = 0$ for some index $i$. Then there exists a subsequence $\mathcal{K}$ such that $\lim_{k \in \mathcal{S}} s_i + \mu^B = 0$. Then all terms except the barrier terms (E) and (G) are bounded below for all $k$. By construction, $(w^E + s^E + \mu^B e) > 0$, and $\{s_k\}$ and $\{w_k\}$ are bounded. Then (E) goes to $\infty$ on $\mathcal{S}$, contradiction lemma

(5.2.1). Similarly, if $\lim_{k\in\mathcal{S}} w_i + \mu^B = 0$, then (G) goes to $\infty$ of $\mathcal{S}$, contradicting the same lemma.

Part 3 follows from parts 1 and 2, and Assumptions 5.2.1 and 5.2.2.

For part 4, it suffices to show that each term in $M$ is bounded below. (A) is bounded below by assumptions 5.2.1 and 5.2.2. (B) is bounded below by the boundedness of $\{x_k\}$ and $\{s_k\}$, and the continuity of $c$. (C) and (D) are bounded below by zero, and the sum of (F), (H), and (I) are bounded below by construction, as $s_k + \mu^B e > 0$ and $w_k + \mu^B e > 0$ for all $k$. The barrier terms are bounded below by the boundedness of $\{s_k\}$ and $\{w_k\}$ established in part 1. Therefore, $M(v_k)$ is bounded below by some $\bar{M}$.

Part 5 follows from parts 1 and 2, Assumptions 5.2.1 and 5.2.2, and the fact that $B^{(i)}$ are constructed to always be positive definite.

Parts 6 and 7 follow from parts 1 and 2, Assumptions 5.2.1 and 5.2.2. □

Lemma 5.2.2 validates Assumptions 2.5.2, 2.5.3, 2.5.4, 2.5.5, 2.5.6, and 2.5.10.

**First-order convergence of Algorithm 5.1**

This section shows the convergence of iterates $\{v_k\}$ generated by Algorithm 5.1 to first-order stationary points of $M$. Let $||\cdot||_{\mathcal{S}}$ denote the norm $||v||_{\mathcal{S}} = \sup_{k\in\mathcal{S}} ||v||_{B_k^{(i)}}$, $||\cdot||_{\mathcal{S}\star}$ its dual, and $||H||_{\mathcal{S}} = ||H||_{\mathcal{S},\mathcal{S}\star}$ the induced matrix norm. The following proofs differ slightly from section 2.5 in that some slight modifications are needed to address the extra slack reset step when an iteration is successful.

**Lemma 5.2.3.** *There exists constants $H_1$ and $H_2$ such that $||\nabla^2 f(v_k)||_{B_k^{(i)}} \leq H_1$ and $||H_k^M||_{B_k^{(i)}} \leq H_2$ for all iterations $k$.*

*Proof.* The result follows directly from Lemma 5.2.2 parts 5, 6, and 7. □

**Theorem 5.2.4.** *For all iterations $k$, it holds that*

$$|M(\widehat{v}_{k+1}) - q_k(\widehat{v}_{k+1})| \leq \max\{H_1, H_2\}\delta_k^2.$$

*Proof.* By the mean value theorem, there exists a $\xi_k \in [0, 1]$ such that

$$M(\widehat{v}_{k+1}) = M(v_k) + \nabla M(v_k)^{\mathrm{T}} d_{v,k} + \frac{1}{2} d_{v,k}^{\mathrm{T}} \nabla^2 M(v_k + \xi_k d_{v,k}) d_{v,k}.$$

Subtracting $q_k(\widehat{v}_{k+1})$ and taking the absolute value yields

$$\begin{aligned}
|M(\widehat{v}_{k+1}) - q_k(\widehat{v}_{k+1})| &= \frac{1}{2}|d_{v,k}^{\mathrm{T}} \nabla^2 M(v_k + \xi_k d_{v,k}) d_{v,k} - d_{v,k}^{\mathrm{T}} H_k^M d_{v,k}| \\
&\leq \frac{1}{2}|d_{v,k}^{\mathrm{T}} \nabla^2 M(v_k + \xi_k d_{v,k}) d_{v,k}| + \frac{1}{2}|d_{v,k}^{\mathrm{T}} H_k^M d_{v,k}| \\
&\leq \frac{1}{2}(H_1 + H_2)||d_{v,k}||_{B_k^{(i)}}^2 \\
&\leq \max\{H_1, H_2\}\delta_k^2
\end{aligned}$$

□

If the scalar $\tau_2$ is set to $\max\{H_1, H_2\}$, then

$$|M(\widehat{v}_{k+1}) - q_k(\widehat{v}_{k+1})| \leq \tau_2 \delta_k^2. \tag{5.24}$$

Thus, the error between the merit and model functions before the slack reset decreases quadratically with the trust-region radius. Intuitively, this implies that as the radius shrinks, the model function becomes a better and better approximation of the merit function. This leads to the following result, which states that if the trust-region radius is sufficiently small at iteration $k$, then the $k$-th iterate will be very successful, assuming that stationarity has yet to be achieved.

**Theorem 5.2.5.** *Suppose that $\nabla M(v_k) \neq 0$, and that*

$$\delta_k \leq \frac{\tau ||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} (1 - \eta_2)}{\tau_2}.$$

*Then $\rho_k \geq \eta_2$, and iteration $k$ is very successful.*

*Proof.* By construction, $0 < \eta_2 < 1$ and $0 < \tau < 1$, and $\tau_2 \leq H_1$. It follows then that

$$\delta_k < \frac{||\nabla M(v_k)||_{(B_k^{(i)})^{-1}}}{H_1}.$$

By assumption,

$$q_k(v_k) - q_k(\widehat{v}_{k+1}) \geq \tau \delta_k ||\nabla M(v_k)||_{(B_k^{(i)})^{-1}}.$$

On the other hand, Theorem 5.2.4 gives

$$|\rho_k - 1| = \left| \frac{M(\widehat{v}_{k+1}) - q_k(\widehat{v}_{k+1})}{q_k(v_k) - q_k(\widehat{v}_{k+1})} \right| \leq \frac{\tau_2}{\tau ||\nabla M(v_k)||_{(B_k^{(i)})^{-1}}} \delta_k \leq 1 - \eta_2.$$

Therefore, iteration $k$ is very successful. $\square$

The following result shows that not only can the trust-region radii not be too small in between two iterates, but that the entire sequence of trust-region radii $\{\delta_k\}$ is bounded below while the optimality conditions fail to hold.

**Theorem 5.2.6.** *Suppose that there exists a constant $\bar{g}$ such that $||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} \geq \bar{g}$ for all iterations $k$ in a subsequence $\mathcal{S}$ of the iterates. Then there exists a constant $\bar{\delta}$ such that $\delta_k \geq \bar{\delta}$ for all iterations $k$.*

*Proof.* The result follows from Theorem 5.2.5, Lemma 5.2.2, and Theorem 2.5.17 $\square$

This result can be used to show that if there are only finitely many successful iterations, then the sequence $\{v_k\}$ generated by Algorithm 5.1 converges to a first-order stationary point of $M$.

**Theorem 5.2.7.** *Suppose that $|\mathcal{S}| < \infty$. Then $v_k = v^*$ for all $k$ sufficiently large, and $v^*$ is a first-order stationary point.*

*Proof.* Suppose that $k$ is the final successful iterate. Then $v_{k+j} = v_{k+1} = v^*$ for all $j > 0$. As all subsequent iterates are unsuccessful, $\delta_k \to 0$. If $v_{k+1}$ is not a stationary point, i.e. $||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} \geq 0$, then 5.2.6 implies that there must be a subsequent successful iteration. This is a contradiction, and thus $||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} = 0$. The result follows. $\square$

Consider the more general case where $\mathcal{S}$ is infinite. The following result shows convergence to a first-order stationary point on a subset of the iterates.

**Theorem 5.2.8.** *It holds that* $\liminf_{k\to\infty} ||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} = 0$.

*Proof.* Suppose there exists an $\varepsilon > 0$ such that $||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} \geq \varepsilon$ for all iterations $k$. Suppose that the $k$-th iterate is successful, so that

$$M(v_k) - M(v_{k+1}) \geq M(v_k) - M(\widehat{v}_{k+1}) \geq \eta_1(q_k(v_k) - q_k(\widehat{v}_{k+1})) \geq \tau\varepsilon\eta_1 \min\left\{\frac{\varepsilon}{H_2}, \bar{\delta}\right\},$$

where $\bar{\delta}$ is the lower bound on the trust-region radii guaranteed by Theorem 5.2.6. Let $n_k$ denote the

number of successful iterations in between the first and $k$-th iterate. It then follows that

$$\begin{aligned} M(v_0) - M(v_{k+1}) &= \sum_{j=0, j\in\mathcal{S}}^{k} (M(v_j) - M(v_{j+1})) \\ &\geq \sum_{j=0, j\in\mathcal{S}}^{k} (M(v_j) - M(\widehat{v}_{j+1})) \\ &\geq n_k \tau\varepsilon\eta_1 \min\left\{\frac{\varepsilon}{H_2}, \bar{\delta}\right\}. \end{aligned}$$

Now, as there are infinitely many successful iterations, $n_k \to \infty$ as $k \to \infty$. Thus, the difference between $M(v_0)$ and $M(v_{k+1})$ is unbounded. This, however, contradicts Lemma 5.2.2 part 4. The result follows. □

Theorem 5.2.8 can be leveraged to prove the following stronger result.

**Theorem 5.2.9.** *It holds that* $\lim_{k\to\infty} ||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} = 0$.

*Proof.* For the sake of establishing a contradiction, assume that there exists a subsequence of successful iterates such that $||\nabla M(v_k)||_{\mathcal{S}\star} \geq 2\varepsilon$ for some $\varepsilon > 0$. Then $||\nabla M(v_k)||_{(B_k^{(i)})^{-1}} \geq 2\varepsilon$. Theorem 5.2.8 and Lemma 5.2.2 part 5 ensure that for each iteration $k$, there exists a subsequent iteration $\ell_k$ such that $\ell_k$ is the first iteration after $k$ to satisfy $||\nabla M(v_{\ell_k})||_{\mathcal{S}\star} \leq ||\nabla M(v_{\ell_k})||_{(B_{\ell_k}^{(i)})^{-1}} < \varepsilon$. Let $c$ be the equivalence constant of the family of the uniformly equivalent norms $||\cdot||_{B_k^{(i)}}$. Let $\mathcal{K}$ denote the subset of successful iterations from $k$ to $\ell_k$, i.e.,

$$\mathcal{K} = \{j \in \mathcal{S} : k \leq j \leq \ell_k\}.$$

Then, it holds that, for all $k \in \mathcal{K}$,

$$\begin{aligned} M(v_k) - M(v_{k+1}) &\geq M(v_k) - M(\widehat{v}_{k+1}) \\ &\geq \eta_1(q_k(v_k) - q_k(\widehat{v}_{k+1})) \\ &\geq \tau\varepsilon\eta_1 \min\left\{\frac{\varepsilon}{H_2}, \delta_k\right\}. \end{aligned} \tag{5.25}$$

Now, the sequence $\{M(v_k)\}$ is monotonically decreasing and bounded below by Lemma 5.2.1 and 5.2.2 part 4, so $M(v_k) - M(v_{k+1}) \to 0$, and therefore $\delta_k \to 0$. It follows then that, for $k$ sufficiently large,

$$\delta_k \leq \frac{1}{\tau\varepsilon\eta_1}(M(v_k) - M(v_{k+1})). \tag{5.26}$$

From Lemma 5.2.2 part 5, it follows that

$$\begin{aligned}||v_k - v_{\ell_k}||_{\mathcal{S}} &\leq \sum_{i=k, i\in\mathcal{K}}^{\ell_k} ||v_i - v_{i+1}||_{\mathcal{S}} \\ &\leq c \sum_{i=k, i\in\mathcal{K}}^{\ell_k} ||v_i - v_{i+1}||_{B_k^{(i)}} \\ &\leq c \sum_{i=k, i\in\mathcal{K}}^{\ell_k} \delta_i \\ &\leq \frac{c}{\tau\varepsilon\eta_1}(M(v_k) - M(v_{k+1})).\end{aligned}$$

The right-hand side of this inequality must converge to zero, and thus $||v_k - v_{\ell_k}||_{\mathcal{S}}$ converges to zero. By the continuity of the gradient of $M$, it follows that $||\nabla M(v_k) - \nabla M(v_{\ell_k})||_{\mathcal{S}\star}$ converges to zero. However, the definitions of $k$ and $\ell_k$ ensure that $||\nabla M(v_k) - \nabla M(v_{\ell_k})||_{\mathcal{S}\star} \geq \varepsilon$. The result follows from this contradiction. □

Thus, iterates generated by Algorithm 5.1 converge to first-order stationary points. Next, convergence to points satisfying second-order optimality conditions is established. The second-order information of the merit function $M$ and the model function $q_k$ must be examined to establish convergence to second-order stationary points.

**Lemma 5.2.10.** *It holds that* $\lim_{k\to\infty} ||\nabla^2 M(v_k) - H_k^M|| = 0$ *whenever* $\lim_{k\to} ||\nabla M(v_k)|| = 0$.

*Proof.* The result follows from the definitions of $\nabla M$, $\nabla^2 M$, and $H_k^M$ combined with Lemma 5.2.2 part 3. □

The above lemma validates assumption 2.5.8. The final result can be directly stated with the last assumption from section 2.5 validated.

**Theorem 5.2.11.** *Let $v^*$ be any limit point of the sequence of iterates generated by Algorithm 5.1, Then $v^*$ satisfies the second-order necessary conditions for optimality.*

*Proof.* The result follows from Theorem 2.5.30, Lemmas 5.2.2 and 5.2.10, and the fact that $M(v_k) \leq M(\widehat{v}_{k+1})$ for all iterations $k$. □

Theorems 5.2.9 and 5.2.11 demonstrate that neither the slack reset nor the lack of uniform continuity on the entire shifted feasible set weaken the results of the basic trust-region algorithm when applied to the merit function $M$. Thus, Algorithm 5.1 is well-suited to guarantee convergence to a minimizer of problem (5.2) along the sequences of inner iterations.

### 5.2.3 Solving the Constrained Nonlinear Optimization Problem

This section discusses and analyzes the outer algorithm used to solve problem (5.2). The algorithm is based on an inner and outer iteration strategy, where the inner iterations use a few steps of Algorithm 5.1 to take steps towards a minimizer of the merit function, and the outer iterations adjust the parameters that were treated as fixed in the previous section.

## The Algorithm

The proposed algorithm is presented in Algorithm 5.2. The method separates iterates into O, M, and F iterates, depending on how the algorithm is coming along.

Optimality iterates, or O iterates, are characterized by a non-negligible improvement to the optimality conditions. This progress is defined in terms of the following measures of the feasibility, stationarity, and complementarity conditions:

$$\begin{aligned} \chi_{\text{feas}}(v_{k+1}) &= ||c(x_{k+1}) - s_{k+1}|| \\ \chi_{\text{stny}}(v_{k+1}) &= \max\{||g(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} y_{k+1}||, ||y_{k+1} - w_{k+1}||\}, \text{ and} \\ \chi_{\text{comp}}(v_{k+1}; \mu_k^B) &= ||\min\{q_1(v_{k+1}), q_2(v_{k+1}; \mu_k^B)\}||, \end{aligned} \tag{5.27}$$

where

$$\begin{aligned} q_1(v_{k+1}) &= \max\{|\min\{s_{k+1}, w_{k+1}, 0\}|, |s_{k+1} \cdot w_{k+1}|\}, \\ q_2(v_{k+1}\ \mu_k^B) &= \max\{\mu_k^B e, |\min\{s_{k+1} + \mu_k^B e, w_{k+1} + \mu_k^B e, 0\}|, |(s_{k+1} + \mu_k^B e) \cdot (w_{k+1} + \mu_k^B e)|\}. \end{aligned}$$

With these definitions in mind, a first-order KKT point of $v_{k+1}$ for problem (5.2) satisfies

$$\chi(v, \mu) = \chi_{\text{feas}}(v) + \chi_{\text{stny}}(v) + \chi_{\text{comp}}(v, \mu) = 0. \tag{5.28}$$

Iteration $k$ is classified as an O-iterate if $\chi(v_{k+1}, \mu_k^B) \leq \chi_k^{\max}$, where $\{\chi_k^{\max}\}$ is a monotonically decreasing positive sequence. At such an iterate, the parameters are updated as $s_{k+1}^E = \max\{0, s_{k+1}\}$, $y_{k+1}^E = y_{k+1}$, $w_{k+1}^E = w_{k+1}$ and $\chi_{k+1}^{\max} = \chi_k^{\max}/2$. This reflects the fact the current iterate $v_{k+1}$ represents the best-known approximation of the optimal values for $s$, $y$, and $w$, and ensures that if there are infinitely many O-iterates, then $\{\chi_k^{\max}\}$ is driven to zero monotonically.

If the test for an O-iterate fails, a second test is used to determine if the current iterate $v_{k+1}$ is

an approximate first-order solution to the unconstrained problem

$$\underset{v=(x,s,y,w)}{\text{minimize}} M(v\ s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B). \tag{5.29}$$

To be more precise, an iterate is called an M-iterate (or a merit iterated), if $v_{k+1}$ satisfies

$$||\nabla_x M(v_{k+1}; s_{k+1}^E, y_{k+1}^E, w_{k+1}^E, \mu^P, \mu^B)||_\infty \leq \tau_k, \tag{5.30a}$$

$$||\nabla_s M(v_{k+1}; s_{k+1}^E, y_{k+1}^E, w_{k+1}^E, \mu^P, \mu^B)||_\infty \leq \tau_k, \tag{5.30b}$$

$$||\nabla_y M(v_{k+1}; s_{k+1}^E, y_{k+1}^E, w_{k+1}^E, \mu^P, \mu^B)||_\infty \leq \tau_k ||D_{k+1}^y||_\infty, \text{ and} \tag{5.30c}$$

$$||\nabla_w M(v_{k+1}; s_{k+1}^E, y_{k+1}^E, w_{k+1}^E, \mu^P, \mu^B)||_\infty \leq \tau_k ||D_{k+1}^w||_\infty, \tag{5.30d}$$

where $\tau_k$ is a positive tolerance. In this case, $v_{k+1}$ is classified as an M-iterate due to the fact that it is an approximate first-order solution of (5.29). In this case, the estimates $s^E$, $y^E$, and $w^E$ are defined by the safeguarded update

$$\begin{aligned} s_{k+1}^E &= \max\{\max\{0, s_{k+1}\}, s_{\max} e\}, \\ y_{k+1}^E &= \max\{-y_{\max} e, \min\{y_{k+1}, y_{\max} e\}\}, \text{ and} \\ w_{k+1}^E &= \max\{w_{k+1}, w_{\max} e\}, \end{aligned} \tag{5.31}$$

where $s_{\max}$, $y_{\max}$, and $w_{\max}^E$ are large positive constants. The method then checks if

$$\chi_{\text{feas}}(v_{k+1}) \leq \tau_k.$$

If so, the penalty parameter $\mu^P$ remains unchanged. Otherwise, it is reduced by $\frac{1}{2}$ to place greater emphasis on satisfying the equality constraints $c(x) - s = 0$ on subsequent iterations. Similarly, the method checks if

$$\chi_{\text{comp}}(v_{k+1}, \mu_k^B) \leq \tau_k,\ s_{k+1} \geq -\tau_k e, \text{ and } w_{k+1} \geq -\tau_k e. \tag{5.32}$$

The barrier parameter $\mu^B$ remains unchanged if these conditions are true. Otherwise, it is reduced by a factor of $1/2$ to place greater emphasis on satisfying the complementarity conditions $s \cdot w = 0$.

An iterate $v_{k+1}$ that cannot be classified as an O or an M iterate is called an F-iterate, or a failed iterate. F-iterates neither improve the optimality of the overall constrained problem nor the solution to the merit function. In this case, all parameters are left unchanged so that progress is measured solely in terms of the reduction of the merit function.

**Algorithm 5.2.** All Shifted Trust-Region Interior Method

Given initial point $v_0 = (x_0^{\mathrm{T}}, s_0^{\mathrm{T}}, y_0^{\mathrm{T}}, w_0^{\mathrm{T}})^{\mathrm{T}}$, where $(s_0, w_0) > 0$.
Given constants $\eta_1, \eta_2, \gamma_C, \gamma_E, \delta_0$ such that $0 < \eta_1 < \eta_2 < 1$, $\eta_1 < 1/2$, $0 < \gamma_C < 1 < \gamma_E$, and $\delta_0 > 0$.
Given constants $y_{\max} > 0$, $w_{\max} > 0$, $s_{\max} > 0$.
Given constants $\mu_0^P > 0$ and $\mu_0^B > 0$.
Choose $w_0^E$ and $s_0^E$ such that $w_0^E + s_0^E + \mu^B e > 0$.
Choose $y_0^E$, $\chi_0^{\max} > 0$.
$k \leftarrow 0$
**while** $||\nabla M(v_k)|| > 0$ **do**
    $(s^E, y^E, w^E, \mu^P, \mu^B) \leftarrow (s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)$
    Compute $v_{k+1}$ in steps 4-19 of Algorithm 5.1 until the sufficient decrease condition is achieved.
    **if** $\chi(v_{k+1}; \mu_k^B) \leq \chi_k^{\max}$ **then** ▷ O-Iterate
        $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\frac{1}{2}\chi_k^{\max}, y_{k+1}, w_{k+1}, \mu_k^P, \mu_k^B, \tau_k)$
        $s_{k+1}^E \leftarrow \max\{0, s_{k+1}\}$
    **else if** $v_{k+1}$ satisfies (5.30 **then** ▷ M-Iterate
        $(\chi_{k+1}^{\max}, \tau_{k+1}) \leftarrow (\chi_k^{\max}, \tau_k)$
        Update $s_{k+1}^E$, $y_{k+1}^E$, $w_{k+1}^E$ using 5.31
        **if** $\chi_{\text{feas}}(v_{k+1}) \leq \tau_k$ **then**
            $\mu_{k+1}^P \leftarrow \mu_k^P$
        **else**
            $\mu_{k+1}^P \leftarrow \mu_k^P/2$
        **end if**
        **if** $\chi_{\text{comp}}(v_{k+1}; \mu_k^B) \leq \tau_k$, $s_{k+1} \geq -\tau_k e$, and $w_{k+1} \geq -\tau_k e$ **then**
            $\mu_{k+1}^B \leftarrow \mu_k^B$
        **else**
            $\mu_{k+1}^B \leftarrow \mu_k^B/2$
            Reset $s_{k+1}$ and $w_{k+1}$ so that $s_{k+1} + \mu^B e > 0$ and $w_{k+1} + \mu^B e > 0$
        **end if**
    **else** ▷ F-Iterate
        $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\chi_k^{\max}, y_k^E, w_k^E, \mu_k^P, \mu_k^B, \tau_k)$
    **end if**
**end while**

Now, the initial shifting of the feasible set by $\mu_0^B$ and subsequent reduction of the barrier parameter implies that, at some M iterations, the reduction of $\mu_{k+1}^B$ may force the current values of $s_{k+1}$ and $w_{k+1}$ to violate the shifted constraints $s_{k+1} + \mu_{k+1}^B e \geq 0$ and $w_{k+1} + \mu_{k+1}^B e \geq 0$. If, for some iteration $k$, a multiplier $[w_k]_i$ becomes infeasible, it is reset as $\max\{y_i, \frac{1}{2}w_i\}$. If a slack variable $[s_k]_i$ is infeasible, the inequality constraint $s_i \geq 0$ is temporarily converted into a trivial equality constraint $s_i = 0$. The primal-dual augmented Lagrangian term enforces this constraint in the merit function (terms (C) and (D) of the merit function), until the constraint value $c(x)_i$ becomes large enough that it reenters the shifted feasible set, at which point $s_i$ is assigned $c(x)_i$. At this point, it becomes free to change value again. The corresponding Lagrange multiplier is reinitialized as $\max\{y_i, \varepsilon\}$, where $\varepsilon$ is some positive constant.

## Convergence Analysis

Convergence of the iterates is established under the properties of the *complementary approximate KKT condition* (CAKKT) proposed by Andreani, Martínez, and Svaiter [2].

**Definition 5.2.1** (CAKKT Condition)**.** A feasible point $(x^*, s^*)$ (i.e., a point such that $c(x^*) - s^* = 0$ and $s^* \geq 0$) is said to satisfy the CAKKT condition if there exists a sequence $\{(x_j, s_j, y_j, w_j)\}$ with $x_j \to x^*$ and $s_j \to s^*$ such that

$$g(x_j) - J(x_j)^{\mathrm{T}} y_j \to 0, \tag{5.33a}$$

$$y_j - z_j \to 0, \tag{5.33b}$$

$$w_j \geq 0, \text{ and} \tag{5.33c}$$

$$s_j \cdot w_j \to 0. \tag{5.33d}$$

Any such $(x^*, s^*)$ is called a CAKKT point.

The CAKKT condition is a sequential optimality condition that holds for each local minimizer. Compared to other sequential optimality conditions, it is relatively tight insofar as not many CAKKT points are not local minimizers. The method for relating CAKKT points with KKT points is given by CAKKT-regularity, which is the weakest known constraint qualification that guarantees the following result.

**Theorem 5.2.12.** *If $(x^*, s^*)$ is a CAKKT point that satisfies CAKKT regularity, then $(x^*, s^*)$ is a first-order KKT point for* (5.2)*.*

The first part of the analysis concerns the conditions under which limit points of the sequence

$\{(x^*, s^*)\}$ are CAKKT points. As the results are tied to the different iterations types, the following index sets are defined:

$$\mathcal{O} = \{k \ : \ \text{iteration k is an O-iteration}\}$$

$$\mathcal{M} = \{k \ : \ \text{iteration k is an M-iteration}\}$$

$$\mathcal{F} = \{k \ : \ \text{iteration k is an F-iteration}\}$$

The first part of the analysis establishes that limit points of the sequence of O-iterates are CAKKT points.

**Lemma 5.2.13.** *If $|\mathcal{O}| = \infty$, then there exists at least one limit point $(x^*, s^*)$ of the infinite sequence $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{O}}$, and any such limit point is a CAKKT point.*

*Proof.* Assumption 5.2.2 implies that there must exist at least one limit point of $\{x_{k+1}\}_{k\in\mathcal{O}}$. If $x^*$ is such a limit point, assumption 5.2.1 implies the existence of a subsequence $\mathcal{K} \subseteq \mathcal{O}$ such that $\{x_{k+1}\}_{k\in\mathcal{K}} \to x^*$ and $\{c(x_{k+1})\}_{k\in\mathcal{K}} \to c(x^*)$. As $|\mathcal{O}| = \infty$, $\{\chi_k^{\max}\} \to 0$. Furthermore, as $\chi(v_{k+1}, \mu_k^B) \leq \chi_k^{\max}$ for all $k \in \mathcal{K}$, and $\chi_{\text{feas}}(v_{k+1}) \leq \chi(v_{k+1}; \mu_k^B)$ for all $k$, it follows that $\chi_{\text{feas}}(v_{k+1})$ converges to zero on $\mathcal{K}$. With the definition $s^* = c(x^*)$, if follows that $\{s_{k+1}\}_{k\in\mathcal{K}} \to \lim_{k\in\mathcal{K}} c(x_{k+1}) = c(x^*) = s^*$, which implies that $(x^*, s^*)$ is feasible for the general constraints because $c(x^*) - s^* = 0$. The remaining feasibility condition is proved componentwise. For any $1 \leq i \leq m$, define

$$Q_1 = \{k \ : \ [q_1(v_{k+1})]_i \leq [q_2(v_{k+1}, \mu_k^B)]_i\} \quad \text{and} \quad Q_2 = \{k \ : \ [q_1(v_{k+1})]_i > [q_2(v_{k+1}, \mu_k^B)]_i\},$$

where $q_1$ and $q_2$ are as they are in the definition of $\chi_{\text{comp}}$. If $\mathcal{K} \cap Q_1$ is infinite, then it follows from the inequalities $\{\chi_{\text{comp}}(v_{k+1}; \mu_k^B)\}_{k\in\mathcal{K}} \leq \{\chi(v_{k+1}, \mu_k^B)\}_{k\in\mathcal{K}} \leq \{\chi_k^{\max}\}_{k\in\mathcal{K}} \to 0$ that $s_i^* = \lim_{k\in\mathcal{K}\cap Q_1}[s_{k+1}]_i \geq 0$. Similarly, if $\mathcal{K} \cap Q_2$ is infinite, then $s_i^* = \lim_{k\in\mathcal{K}\cap Q_2}[s_{k+1}]_i = \lim_{k\in\mathcal{K}\cap Q_2}[s_{k+1} + \mu^B e]_i \geq 0$, where the second inequality uses the limit $\{\mu_k^B\} \to 0$ that follows from the definition of $Q_2$. Combining these two cases implies $s_i^* \geq 0$, as claimed. It follows that the limit point $(x^*, s^*)$ is feasible.

It remains to show that $(x^*, s^*)$ is a CAKKT point. Let

$$[\bar{s}_{k+1}]_i = \begin{cases} [s_{k+1}]_i & \text{if } k \in Q_1 \\ [s_{k+1}]_i + \mu^B & \text{if } k \in Q_2 \end{cases}$$

and

$$[\bar{w}_{k+1}]_i = \begin{cases} \max\{[w_{k+1}]_i, 0\} & \text{if } k \in Q_1 \\ [w_{k+1}]_i + \mu^B & \text{if } k \in Q_2 \end{cases}$$

for every $1 \leq i \leq m$, and consider the sequence $(x_{k+1}, \bar{s}_{k+1}, y_{k+1}, \bar{w}_{k+1})$ as a candidate for the sequence used in Definition 5.2.1 to verify that $(s^*, s^*)$ is a CAKKT point. If $\mathcal{O} \cap Q_2$ is finite, then it follows from the definition of $\bar{s}_{k+1}$ and the limit $s_{k+1} \to s^*$ that $\{[\bar{s}_{k+1}]_i\}_{k \in \mathcal{K}} \to s_i^\star$. Furthermore, $\{\chi_{\text{comp}}(v_{k+1;\mu_k^B})\}_{k \in \mathcal{K}} \to 0$ implies that $\liminf_{k \in \mathcal{K}} [w_{k+1}] \geq 0$, and therefore $\{[\bar{w}_{k+1} - w_{k+1}]_i\}_{k \in \mathcal{K}} \to 0$. On the other hand, if $\mathcal{O} \cap Q_2$ is infinite, then the definitions of $Q_2$ and $\chi_{\text{comp}}(v_{k+1}; \mu_k^B)$, together with the limit $\{\chi_{\text{comp}}(v_{k+1}, \mu_k^B)\}_{k \in \mathcal{K}} \to 0$ imply that $\{\mu_k^B\} \to 0$, giving $\{[\bar{s}_{k+1}]_i\}_{k \in \mathcal{K}} \to s_i^*$ and $\{[\bar{w}_{k+1} - w_{k+1}]_i\}_{k \in \mathcal{K}} \to 0$. A $i$ is an arbitrary index, these cases together imply that $\{\bar{s}_{k+1}\}_{k \in \mathcal{K}} \to s^*$ and $\{\bar{w}_{k+1} - w_{k+1}\}_{k \in \mathcal{K}} \to 0$.

The next step is to show that $\{(x_{k+1}, \bar{s}_{k+1}, y_{k+1}, \bar{w}_{k+1})\}_{k \in \mathcal{K}}$ satisfies the conditions required by Definition 5.2.1. It follows from the limit $\{\chi(v_{k+1}\ \mu_k^B)\}_{k \in \mathcal{K}} \to 0$ established above that $\{\chi_{\text{stny}}(v_{k+1}) + \chi_{\text{comp}}(v_{k+1}; \mu_k^B)\}_{k \in \mathcal{K}} \to 0$. This, together with the limit $\{\bar{w}_{k+1} - w_{k+1}\}_{k \in \mathcal{K}} \to 0$ implies that $\{\nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} y_{k+1}\}_{k \in \mathcal{K}} \to 0$ and $\{y_{k+1} - w_{k+1}\} \to 0$, which establishes that the first two conditions of Definition 5.2.1 hold. Then nonnegativity of $\bar{w}_{k+1}$ for all $k$ is clear from the definition. It must finally be shown that the sequential complementarity holds. Consider the $i$-th components of $\bar{w}_{k+1}$ and $\bar{w}_{k+1}$. If the set $\mathcal{K} \cap Q_1$ is infinite, then the definitions of $\bar{s}_{k+1}$, $q_2(v_{k+1}, \mu_k^B)$, and $\chi_{\text{comp}}(v_{k+1}, \mu_k^B)$, along with the limit $\{\chi_{\text{comp}}(v_{k+1}; \mu_k^B)\}_{k \in \mathcal{K}} \to 0$ imply that $\{[\bar{w}_{k+1} \cdot \bar{s}_{k+1}]_i\}_{k \in \mathcal{K} \cap Q_1} \to 0$. On the other hand, if the set $\mathcal{K} \cap Q_2$ is infinite, then the definitions of $\bar{s}_{k+1}$, $q_2(v_{k+1}, \mu_k^B)$, and $\chi_{\text{comp}}(v_{k+1}, \mu_k^B)$, along with the limits $\{\chi_{\text{comp}}(v_{k+1}; \mu_k^B)\}_{k \in \mathcal{K}} \to 0$ and $\{\bar{w}_{k+1} - w_{k+1}\}_{k \in \mathcal{K}} \to 0$ imply that $\{[\bar{w}_{k+1} \cdot \bar{s}_{k+1}]_i\}_{k \in \mathcal{K} \cap Q_2} \to 0$. Combining these two shows that the last condition is satisfied. Therefore, $(x^*, s^*)$ is a CAKKT point. $\square$

In the complementary case, where $|\mathcal{O}| < \infty$, it will be shown that every limit point of the iterations sequence $\{x_{k+1}, s_{k+1}\}_{k \in \mathcal{M}}$ is infeasible with respect to the constraint $c(x) - s = 0$, but does solve the least-infeasibility problem

$$\underset{x \in \mathbb{R}^n, s \in \mathbb{R}^m}{\text{minimize}} \frac{1}{2} ||c(x) - s||_2^2 \quad \text{subject to} \quad s \geq 0. \tag{5.34}$$

The first-order KKT conditions to (5.34) are

$$J(x)^{\mathrm{T}}(c(x) - s) = 0, \qquad s^* \geq 0, \tag{5.35a}$$

$$s \cdot (c(x) - s) = 0, \qquad c(x) - s \leq 0. \tag{5.35b}$$

These conditions define an *infeasible stationary point.*

**Definition 5.2.2** (Infeasible Stationary Point)**.** The pair $(x^*, s^*)$ is an *infeasible stationary point* if $c(x^*) - s^* \neq 0$ and $(x^*, s^*)$ satisfies conditions (5.35).

The following result shows that if there are only finitely many O-iterations, then there are infinitely many M-iterations.

**Lemma 5.2.14.** *If $|\mathcal{O}| < \infty$, then $|\mathcal{M}| = \infty$.*

*Proof.* Suppose that $|\mathcal{M}| < \infty$, in which case $|\mathcal{O} \cup \mathcal{M}| < \infty$. Therefore, $k \in \mathcal{F}$ for all $k$ sufficiently large, i.e., there exists an iteration index $k_F$ such that

$$k \in \mathcal{F},\ y_k^E = y_k,\ \text{and}\ (\tau_k, w_k^E, \mu_k^P, \mu_k^B) = (\tau, w^E, \mu^P, \mu^B) > 0$$

for all $k \geq k_F$. This means that the iterates computed by Algorithm 5.2 are equivalent to the successful iterations of 5.1 for all $k \geq k_F$. In this case, Lemma 5.2.2 and Theorem 5.2.9 can be applied to show that the (5.30) is satisfied for all $k$ sufficiently large. This contradicts the assumption that the number of M-iterates is finite. The result follows. □

The following result justifies the right-hand side of (5.30).

**Lemma 5.2.15.** *If $|\mathcal{M}| = \infty$ then*

$$\begin{aligned}
\lim_{k \in \mathcal{M}} ||\pi_{k+1}^y - y_{k+1}||_\infty &= 0, \\
\lim_{k \in \mathcal{M}} ||\pi_{k+1}^w - w_{k+1}||_\infty &= 0, \\
\lim_{k \in \mathcal{M}} ||\pi_{k+1}^y - \pi_{k+1}^w||_\infty &= 0,\ \textit{and} \\
\lim_{k \in \mathcal{M}} ||y_{k+1} - w_{k+1}||_\infty &= 0,
\end{aligned}$$

*Proof.* It follows from (5.9) and (5.30) that

$$||\pi_{k+1}^y - y_{k+1}||_\infty \leq \tau_k, \quad \text{and} \quad ||\pi_{k+1}^w - w_{k+1}||_\infty \leq \tau_k. \tag{5.36}$$

By assumption, $|\mathcal{M}| = \infty$, thus $\tau_k \to 0$ as $k \to \infty$. This fact, together with (5.36), establishes the first two limits in the result. It also then follows, by (5.30) and (5.9), that the third limit holds. Finally, it follows that

$$\begin{aligned}
0 &= \lim_{k \in \mathcal{M}} ||\pi_{k+1}^y - \pi_{k+1}^w||_\infty \\
&= \lim_{k \in \mathcal{M}} ||(\pi_{k+1}^y - y_{k+1}) + (y_{k+1} - w_{k+1}) + (w_{k+1} + \pi_{k+1}^w)||_\infty \\
&= \lim_{k \in \mathcal{M}} ||y_{k+1} - w_{k+1}||_\infty.
\end{aligned}$$

□

The following result shows that if the set of O-iterates is finite, then any limit point of the sequence $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{M}}$ is infeasible with respect to $(c(x) - s) = 0$. It is important to note here that an M-iterate satisfies conditions (5.30) and fails to be an O-iterate.

**Lemma 5.2.16.** *If $|\mathcal{O}| < \infty$, then every limit point $(x^*, s^*)$ of the subsequence $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{M}}$ satisfies $c(x^*) - s^* \neq 0$.*

*Proof.* Let $(x^*, s^*)$ be a limit point of the infinite sequence $\mathcal{M}$. Then a subsequence $\mathcal{K} \subseteq \mathcal{M}$ exists on which iterates converge to $(x^*, s^*)$. For the sake of establishing a contradiction, assume that $c(x^*) - s^* = 0$, so that

$$\lim_{k\in\mathcal{K}} ||c(x_{k+1}) - s|| = 0. \tag{5.37}$$

First, it is shown that $s^* \geq 0$, which will imply that $(x^*, s^*)$ is feasible. The trust-region method ensures that $s_{k+1} + \mu_B^k e > 0$ for all $k$. Thus, if $\lim_{k\to\infty} \mu_k^B = 0$, then $s^* \geq 0$. On the other hand, if $\mu_k^B$ does not converge to zero, then that must mean that $\mu^B$ is reduced only a finite number of times so that $\mu_k^B = \mu^B > 0$ and (5.32) holds for all $k \in \mathcal{M}$ sufficiently large. Taking the limit over all $k \in \mathcal{M}$ in (5.32) and recalling that $\lim_{k\to\infty} \tau_k = 0$ gives $s^* \geq 0$. Therefore, the limit point $(x^*, s^*)$ is feasible.

Now the assumption that $\mathcal{O}$ is finite and the implication that $\mathcal{M}$ is infinite, along with the update rule for $\chi_k^{\max}$ establishes that $\lim_{k\to 0} \tau_k = 0$ and

$$\chi_k^{\max} = \chi^{\max} > 0 \text{ for all sufficiently large } k \in \mathcal{K}. \tag{5.38}$$

Using $|\mathcal{O}| < \infty$, Lemma 5.2.15, and the fact that $M$ is defined to be infinite outside the shifted feasible set in Algorithm 5.2 gives

$$\lim_{k\in\mathcal{K}} ||y_{k+1} - w_{k+1}|| = 0, \quad \text{and} \quad w_{k+1} + \mu_{k+1}^B e > 0 \text{ for all } k \geq 0. \tag{5.39}$$

From the definitions of $\pi_{k+1}^y$ and $\nabla_x M$, it holds that

$$\begin{aligned}\nabla f(x_{k+1} - J(x_{k+1}))^{\mathrm{T}} y_{k+1} &= \nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}}(2\pi_{k+1}^y + y_{k+1} - 2\pi_{k+1}^y) \\ &= \nabla_x M(x_{k+1}; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B) - 2J(x_{k+1})^{\mathrm{T}}(y_{k+1} - \pi_{k+1}^y).\end{aligned}$$

Thus,

$$\lim_{k\in\mathcal{K}} \nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} y_{k+1} = 0. \tag{5.40}$$

It is now shown that $\lim_{k\in\mathcal{K}} \chi_{\mathrm{comp}}(v_{k+1}; \mu_k^B) = 0$, the proof of which involves two cases.

**Case 1:** Assume that $\lim_{k\to\infty} \mu_k^B \neq 0$. In this case, $\mu_k^B > 0$ for all sufficiently large $k$. Combined with the fact that $\lim_{k\to\infty} \tau_k = 0$, it must be that (5.32) holds for all $k \in \mathcal{K}$ sufficiently large. Therefore, $\lim k \in \mathcal{K} \chi_{\text{comp}}(v_{k+1}; \mu_k^B) = 0$.

**Case 2:** Assume that $\lim_{k\to\infty} \mu_k^B = 0$. Lemma 5.2.15 implies that $\lim_{k\in\mathcal{K}}(\pi_{k+1}^w - w_{k+1}) = 0$. Now, the matrix sequence $\{S_{k+1} + \mu_k^B I\}_{k\in\mathcal{K}}$ must be bounded, as $s_{k+1} \to s^*$ on $\mathcal{K}$ and $\mu_k^B$ is monotonically decreasing to zero. The definition of $\pi_{k+1}^w$ then gives

$$0 = \lim_{k\in\mathcal{K}} (S_{k+1} + \mu_k^B I)(\pi_{k+1}^w - w_{k+1}) = \lim_{k\in\mathcal{K}} (\mu_k^B w_k^E - (S_{k+1+\mu_k^B I})w_{k+1}). \tag{5.41}$$

Moreover, as $|\mathcal{O}| < \infty$ and $w_k + \mu_k^B > 0$ by construction, the updating strategy for $w_k^E$ in Algorithm 5.2 guarantees that $w_k^E$ is bounded over all iterations $k$. It then follows from (5.41), the uniform boundedness of $\{w_k^E\}$, and the fact that $\lim_{k\to\infty} \mu_k^B = 0$ that

$$0 = \lim_{k\in\mathcal{K}} ([s_{k+1}]_i + \mu_k^B)[w_{k+1}]_i = \lim_{k\in\mathcal{K}} ([s_{k+1}]_i + \mu_k^B)([w_{k+1}]_i + \mu_k^B) \tag{5.42}$$

for all indices $i$. Consider an individual constraint $s_i \geq 0$. This constraint may either be active, in which case $s_i = 0$, or inactive, in which case $s_i > 0$. The two cases are considered separately.

**Subcase 2a:** $s_i^\star > 0$ for some index $i$. As $\lim_{k\in\mathcal{K}}[s_{k+1}]_i = s_i^\star$ and $\lim_{k\to\infty} \mu_k^B = 0$, it follows from (5.42) that $\lim_{k\in\mathcal{K}}[w_{k+1}]_i = 0$. Combining these two limits shows that $\lim_{k\in\mathcal{K}}[q_1(v_{k+1})]_i = 0$.

**Subcase 2b:** $s_i^\star = 0$ for some index $i$. In this case, it follows from $\lim_{k\to\infty} \mu_k^B = 0$, (5.42), the fact that $w_{k+1} + \mu_k^B > 0$, and the limit $\lim_{k\in\mathcal{K}}[s_{k+1}]_i = s_i^\star = 0$ that $\lim_{k\in\mathcal{K}}[q_2(v_{k+1}; \mu_k^B)]_i = 0$.

As one of these two subcases must occur for each index $i$, it follows that

$$\lim_{k\in\mathcal{K}} \chi_{\text{comp}}(v_{k+1}; \mu_k^B) = 0,$$

which completes the proof of case 2.

It has been shown that the limit $\lim_{k\in\mathcal{K}} \chi(v_{k+1}; \mu_k^B) = 0$ holds under the assumption that $c(x^*) - s^* = 0$. Therefore, $k \in \mathcal{O}$ for all $k \in \mathcal{K}$ sufficiently large. However, this contradicts the fact that $|\mathcal{O}| < \infty$, which establishes the result that $c(x^*) - s^*/neq0$. □

The following result establishes that if the number of O-iterates is finite, then the sequence of M-iterates $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{M}}$ has at least one limit point and that any such limit point is an infeasible stationary point.

**Lemma 5.2.17.** *If $|\mathcal{O}| < \infty$, then there exists at least one limit point $(x^*, s^*)$ of the necessarily infinite sequence $\{(x_{k+}, s_{k+1})\}_{k\in\mathcal{M}}$, and any such point is an infeasible stationary point as defined by Definition 5.2.2.*

*Proof.* Suppose that $|\mathcal{O}| < \infty$. Then Lemma 5.2.14 implies that $|\mathcal{M}| = \infty$. The updating strategy of Algorithm 5.2 ensures that $\{s_k^E\}$, $\{y_k^E\}$, and $\{w_k^E\}$ are bounded. The next step is to show that $\{s_{k+1}\}_{k\in\mathcal{M}}$ is bounded as well.

For a proof by contradiction, suppose that there is an index $i$ and a subsequence $\mathcal{K} \subseteq \mathcal{M}$ on which $\lim_{k\in\mathcal{K}}[s_{k+1}]_i = \infty$. When Assumption 5.2.1 and 5.2.2 hold, the sequences $\{c(x_{k+1})\}_{k\in\mathcal{K}}$, $\{\nabla f(x_{k+1})\}_{k\in\mathcal{K}}$, and $\{J(x_{k+1})\}_{k\in\mathcal{K}}$ must be bounded. Therefore, $\{\pi_{k+1}^y\}_{k\in\mathcal{K}}$ must be unbounded. On the other hand, (5.9), (5.30a), the fact that $\lim_{k\to\infty}\tau_k = 0$, and Lemma 5.2.15 imply that

$$\begin{aligned} 0 &= \lim_{k\in\mathcal{M}} ||\nabla_x M(v_{k+1}; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)|| \\ &= \lim_{k\in\mathcal{M}} ||\nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}}\pi_{k+1}^y - J(x_{k+1})^{\mathrm{T}}(\pi_{k+1}^y - y_{k+1})|| \\ &= \lim_{k\in\mathcal{M}} ||\nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}}\pi_{k+1}^y||. \end{aligned}$$

However, this contradicts that $\pi_{k+1}^y$ is unbounded. Therefore, $\{s_{k+1}\}_{k\in\mathcal{M}}$ is bounded.

The next step is establishing the feasibility of $s^*$, i.e., that $s^* \geq 0$. The M-iterate check in Algorithm 5.2, i.e., the check that (5.32) holds, is performed infinitely many times. If (5.32) is satisfied only a finite number of times, then the reduction of $\mu_k^B$ forces $\lim_{k\to\infty}\mu_k^B = 0$. By construction, $s_{k+1} + \mu_k^B > 0$, and therefore $s^* \geq 0$. On the other hand, if (5.32) holds for all but finitely many iterations, then$\mu_{k+1}^B = \mu^B > 0$ for all $k$ sufficiently large. Therefore, $\lim_{k\in\mathcal{K}}\chi_{\text{comp}}(v_{k+1}; \mu_k^B) = 0$, as $\tau_k \to 0$. It follows then that $s^* \geq 0$.

The boundedness of $\{s_{k+1}\}_{k\in\mathcal{M}}$ and the assumed boundedness of $\{x_{k+1}\}_{k\in\mathcal{M}}$ ensures the existence of at least one limit point of $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{M}}$. Let $(x^*, s^*)$ be such a limit point. Then there exists a subsequence $\mathcal{K} \subseteq \mathcal{M}$ such that $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{K}} \to (x^*, s^*)$. It remains to show that $(x^*, s^*)$ is an infeasible stationary point.

As $|\mathcal{O}| < \infty$, it follows from Lemma 5.2.16 that $c(x^*) - s^* \neq 0$. As $\mathcal{K} \subseteq M$ is infinite, it holds that $\lim_{k\to\infty}\tau_k = 0$. It follows then that $\chi_{\text{feas}} \leq \tau_k$ will not hold for all sufficiently large $k \in \mathcal{K}$. The barrier parameter updates ensure $\{\mu_k^P\} \to 0$. Combining this with the boundedness of $\{y_k^E\}$ and Lemma 5.2.15 gives

$$\{c(x_{k+1} - s_{k+1})\}_{k\in\mathcal{K}} \leq \{\mu_k^P(y_k^E + \frac{1}{2}(w_{k+1} - y_{k+1}))\}_{k\in\mathcal{K}} \to 0.$$

This implies that $c(x^*) - s^* \leq 0$, and the second condition in (5.35b) holds.

For a proof of the first condition of (5.35a), observe that the merit function gradients must satisfy $\{\nabla_x M(v_{k+1}; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\}_{k\in\mathcal{K}} \to 0$, as condition (5.30a) is satisfied for all $k \in \mathcal{M}$. Multiplying this by the penalty parameter $\mu_k^P$ and applying the definition of $\pi_{k+1}^y$ gives

$$\lim_{k\in\mathcal{K}} \mu_k^P g(x_{k+1}) - J(x_{k+1})^{\mathrm{T}}(\mu_k^P \pi_{k+1}^y + \mu_k^P(\pi_{k+1}^y - y_{k+1})) \to 0.$$

Combining this with $\lim_{k\in\mathcal{K}} x_{k+1} \to x^*$, $\lim_{k\to\infty} \mu_k^P = 0$, and Lemma 5.2.15 yields

$$\lim_{k\in\mathcal{K}} -J(x_{k+1})^{\mathrm{T}}(\mu_k^P \pi_{k+1}^y) = \lim_{k\in\mathcal{K}} -J(x_{k+1})^{\mathrm{T}}(\mu_k^P y_k^E - (c(x_{k+1}) - s_{k+1})) = 0.$$

The boundedness of $\{y_k^E\}$, and the facts that $\mu_k^P \to 0$ and $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{K}} \to (x^*, s^*)$ shows that the equality condition of (5.35a) holds.

It remains to show that the complementarity condition of (5.35b) holds. Lemma 5.2.15 shows that $\lim_{k\in\mathcal{K}} \pi_{k+1}^w - \pi_{k+1}^y = 0$. Therefore, multiplying the sequence of vectors $\{\pi_{k+1}^w - \pi_{k+1}^y\}$ term by term by the bounded sequence $\{\mu_k^P(S_{k+1} + \mu_k^B I)\}$ does not change the limit. This yields

$$\lim_{k\in\mathcal{K}} \mu_k^B \mu_k^P (w_k^E - s_{k+1} + s_k^E) - \mu_k^P (S_{k+1} + \mu_k^B I) y_k^E + (S_{k+1} + \mu_k^B I)(c(x_{k+1}) - s_{k+1}) \to 0.$$

Thus,

$$\lim_{k\in\mathcal{K}} (S_{k+1} + \mu_k^B I)(c(x_{k+1}) - s_{k+1}) \to 0.$$

As $c(x^*) - s^* \neq 0$, there must exist an index $i$ such that $[c(x^*) - s^*]_i \neq 0$. For this index, it follows that $\lim_{k\in\mathcal{K}} [s_{k+1}]_i + \mu_k^B = 0$. As $s^* \geq 0$, it then follows that $\mu_k^B \to 0$. It then follows that $s^* \cdot (c(x^*) - s^*) = 0$. Thus, all conditions of (5.35) hold, completing the proof. □

The following result directly follows from the preceding discussion.

**Theorem 5.2.18.** *Under Assumption 5.2.1 and 5.2.2, one of the following occurs:*

1. $|\mathcal{O}| = \infty$, *in which case limit points of* $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{O}}$ *exist, and every such limit point* $(x^*, s^*)$ *is a CAKKT point for problem* (5.2). *If, in addition, CAKKT regularity holds at* $(x^*, s^*)$, *then* $(x^*, s^*)$ *is a KKT point for problem* (5.2).

2. $|\mathcal{O}| < \infty$, *in which case* $|\mathcal{M}| = \infty$, *limit points of* $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{M}}$ *exist, and every such limit point* $(x^*, s^*)$ *is an infeasible stationary point of* (5.2).

### 5.2.4 Implementation Details

Numerical results are given for a Fortran implementation of the all-shifted primal-dual penalty-barrier trust-region algorithm (Algorithm 5.2). Results are obtained for the CUTEst NLP test collection (see Bongartz et al. [4] and Gould, Orban, and Toint [19]). The CUTEst problems are given in the form

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{pmatrix} l_x \\ l_s \end{pmatrix} \leq \begin{pmatrix} x \\ c(x) \end{pmatrix} \leq \begin{pmatrix} u_x \\ u_s \end{pmatrix}.$$

The $i$-th constraint is an equality constraint if $[l_x]_i = [u_x]_i$ or $[l_s]_i = [u_s]_i$. A variable or constraint is unbounded either above or below if the bound is greater than or equal to $10^{20}$. Internally, the implementation converts problems to the equivalent form

$$\begin{aligned} &\underset{x\in\mathbb{R}^n, s\in\mathbb{R}^m}{\text{minimize}} \quad f(x) \\ &\text{subject to} \quad c(x) - s = 0, \qquad E_x x = b, \quad E_l x - l_x \geq 0, \quad u_x - E_u x \geq 0, \\ &\qquad\qquad\qquad\qquad\qquad\qquad F_x s = h, \quad F_l s - l_s \geq 0, \quad u_s - F_u s \geq 0. \end{aligned}$$

The matrices $E_x$, $E_l$, $E_u$, $F_x, F_l, F_u$ correspond to matrices with rows of the identity matrix which selects the entries of $x$ and $s$ that are fixed, bounded below, and bounded above, respectively. The constraint $E_x x = b$ is not treated as an equality constraint. Instead, $E_x x$ is held constant, and those particular variables are not allowed to change. Thus, the problem can be written in the equivalent form

$$\begin{aligned} &\underset{x\in\mathbb{R}^n, s\in\mathbb{R}^m}{\text{minimize}} \quad f(x) \\ &\text{subject to} \quad c(x) - s = 0, \quad E_l x - l_x \geq 0, \quad u_x - E_u x \geq 0, \\ &\qquad\qquad\qquad F_x s = h, \quad F_l s - l_s \geq 0, \quad u_s - F_u s \geq 0. \end{aligned}$$

The dual variables corresponding to $E_l x - l_x \geq 0$ and $u_x - E_u x \geq 0$ are denoted as $z_l$ and $z_u$, while the dual variables corresponding to $F_x s = h$, $F_l s - l_s \geq 0$, and $u_s - F_u s \geq 0$ are denoted as $w_x$, $w_l$, and $w_u$. As the algorithm proceeds, any of the variables may become infeasible with respect to the shifted inequality constraints. An infeasible slack variable is treated by temporarily fixing it on its bound, and treating the corresponding constraint function $c(x)$ as an equality constraint. A different strategy is needed should a variable $[x]_i$ become infeasible. Suppose that $[x]_i$ becomes less than $[l_x]_i - \mu^B$. The constraint $[x]_i - [l_x]_i$ is then included as a temporary equality constraint, with its own term in the merit function, and its own Lagrange multiplier $[v_l]_i$. The same is done if $[x]_i$ becomes infeasible with

respect to its upper bound. While it is infeasible, its associated barrier terms are removed from the merit function. Analogously, it is possible for a dual variable to become infeasible with respect to the shifted dual inequality constraints. Suppose $[E_l^{\mathrm{T}} z_l]_i$ becomes infeasible. Then it is simply reinitialized as $\max\{[\nabla f(x) - J(x)^{\mathrm{T}} y - E_u^{\mathrm{T}} z_u]_i, \frac{1}{2}[E_l^{\mathrm{T}} z_l]_i\}$. Similarly, if $[F_l^{\mathrm{T}} w_l]_i$ becomes infeasible, it is reinitialized to $max\{[y - F_x^{\mathrm{T}} w_x - F_u^{\mathrm{T}} w_u]_i, \frac{1}{2}[F_l^{\mathrm{T}} w_l]_i\}$

The iterates are terminated at the first point which satisfies $e_p < \tau_p$ and $e_d < \tau_d$, where $e_p$ and $e_d$ are the primal and dual infeasibilities

$$e_p(x,s) = \left|\left| \begin{pmatrix} \min\{0, E_l x - l_x, u_x - E_u x\} \\ \min\{0, F_l s - l_s, u_s - F_u s\} \\ ||F_x s - h||_\infty \\ ||c(x) - s||_\infty / \max\{1, ||s||_\infty\} \end{pmatrix} \right|\right|_\infty,$$

and

$$e_d(x,s,y,z_l,z_u,w_x,w_l,w_u) = \left|\left| \begin{pmatrix} ||\nabla f(x) - J(x)^{\mathrm{T}} y - E_l^{\mathrm{T}} z_l - E_u^{\mathrm{T}} z_u||_\infty / \sigma \\ ||y - F_x^{\mathrm{T}} w_x - F_l^{\mathrm{T}} w_l - F_u^{\mathrm{T}} w_u||_\infty \\ z_l \cdot \min\{1, E_l x - l_x\} \\ z_u \cdot \min\{1, u_x - E_u x\} \\ w_l \cdot \min\{1, F_l s - l_s\} \\ w_u \cdot \min\{1, u_s - F_u s\} \end{pmatrix} \right|\right|_\infty,$$

where $\sigma = \max\{1, ||\nabla f(x)||_\infty, \max\{1, ||y||_\infty\}||J(x)||_\infty\}$.

Experiments reveal that the basic trust-region approach of Algorithm 5.1 is not always effective. In [17], a flexible projected line-search approach is used to minimize the merit function $M$. Experiments reveal that a similar approach adapted to the basic trust-region algorithm can yield better results. In the flexible line-search strategy, a second barrier parameter $\mu^L$ such that $\mu^L > \mu^P$ is established. Let $F(v;\mu^P)$ denote the path-following equations (5.15) with penalty parameter $\mu^P$, and let $M(v;\mu^P)$ denote the value of the merit function $M$ evaluated at $v$ with barrier parameter $\mu^P$. The sufficient-decrease condition in the flexible trust-region approach is satisfied if

$$M(v_k + d_k; \mu^P) < \max\left\{M(v_k;\mu^P), M_{\max}\right\}, \tag{5.43a}$$

$$M(v_k + d_k; \mu^L) < \max\left\{M(v_k;\mu^L), M_{\max}\right\}, \quad \text{and} \tag{5.43b}$$

$$||F(v_k + d_k; \mu^P)||_\infty < \eta_F \max\left\{||F(v_k + d_k;\mu^P)||_\infty, F_{\max}\right\}, \tag{5.43c}$$

or if

$$M(v_k;\mu^F) - M(v_k + d_k;\mu^F) \geq \eta_1 Q_k(d_k), \tag{5.44}$$

where $\mu^F \in [\mu^P, \mu^L]$, $\eta_F < 1$, and $M_{\max}$ and $F_{\max}$ are large preassigned parameters. If equations (5.43) hold, then $F_{\max}$ is updated to $\eta_F F_{\max}$. It is not feasible to check for all $\mu^F \in [\mu^P, \mu^L]$, so instead, the search is restricted to the two endpoints. If either of these two sets of conditions holds for $\eta_1$ and $\eta_{F,1}$, then the iteration is successful. If they hold for $\eta_2$ and $\eta_{F,2} < \eta_{F,1}$, then it is very successful. With these modifications, Algorithm 5.1 becomes Algorithm 5.3.

**Algorithm 5.3.** Merit Function Flexible Trust-Region Algorithm

Given constants $\eta_1$, $\eta_2$,$\eta_{F,1}$, $\eta_{F,2}$, $\gamma_C$, $\gamma_E$, $\delta_0$ such that $0 < \eta_1 < \eta_2 < 1$, $\eta_1 < 1/2$, $\eta_{F,2} < \eta_{F,1}$, $0 < \gamma_C < 1 < \gamma_E$, and $\delta_0 > 0$
$k \leftarrow 0$
**while** not converged **do**
  $d_k = \operatorname{argmin}_{p\in\mathbb{R}^n}\{Q_k(d_v) \ : \ ||d_v||_{B_k^{(i)}} \leq \delta_k\}$
  **if** (5.43) or (5.44) hold for $\eta_1$ and $\eta_{F,1}$ **then**
    $\widehat{v}_{k+1} = v_k + d_k$
    **if** (5.43) or (5.44) hold for $\eta_2$ and $\eta_{F,2}$ **then**
      $\delta_{k+1} \leftarrow \max\{\delta_k, \gamma_E||d_k||_{B_k^{(i)}}\}$
    **else**
      $\delta_{k+1} \leftarrow \delta_k$
    **end if**
    $s_{k+1} \leftarrow \max\{\widehat{s}_{k+1}, c(x_{k+1}) - \mu^F(y^E + \frac{1}{2}(w_{k+1} - y_{k+1}) + \mu^B e)$ ▷ Slack Reset
    $v_{k+1} \leftarrow (\widehat{x}_{k+1}, s_{k+1}, \widehat{y}_{k+1}, \widehat{w}_{k+1})$
  **else**
    $v_{k+1} \leftarrow v_k$
    $\delta_{k+1} \leftarrow \gamma_C||d_k||_{B_k^{(i)}}$
  **end if**
  $k \leftarrow k + 1$
**end while**

Additionally, Algorithm 5.2 needs to be modified to include an update to $\mu^L$ that ensures that $\mu_k^L > \mu_k^P$ for all $k$. Consider the update

$$\mu_{k+1}^L = \begin{cases} \mu_k^L & \text{if } \ M(v_k;\mu_k^L) - M(v_k + d_k;\mu_k^L) \geq \eta_1 Q_k(d_k) \ \text{ and } \ \mu_{k+1}^P = \mu_k^P, \\ \max\{\frac{1}{2}\mu_k^L, \mu^P\} & \text{otherwise } . \end{cases} \tag{5.45}$$

Algorithm 5.2 then becomes Algorithm 5.4.

Another issue that is important to address is what occurs when computed iterates become arbitrarily close to the shifted boundary. Consider the constraint $x_i \geq 0$ for some index $i$, which, when shifted, becomes $x_i > -\mu^B$. In the theoretical discussion, there are no issues in saying that $x_i$ becomes

**Algorithm 5.4.** All Shifted Flexible Trust-Region Interior Method

Given initial point $v_0 = (x_0, s_0, y_0, w_0)$, where $(s_0, w_0) > 0$.
Given constants $\eta_1, \eta_2, \eta_{F,1}, \eta_{F,2}, \gamma_C, \gamma_E, \delta_0$ such that $0 < \eta_1 < \eta_2 < 1$, $\eta_1 < 1/2$, $\eta_{F,2} < \eta_{F,1}$, $0 < \gamma_C < 1 < \gamma_E$, and $\delta_0 > 0$.
Given constants $y_{\max} > 0$, $w_{\max} > 0$, $s_{\max} > 0$.
Given constants $\mu_0^P > 0$ and $\mu_0^B > 0$.
Choose $w_0^E$ and $s_0^E$ such that $w_0^E + s_0^E + \mu^B e > 0$.
Choose $y_0^E$, $\chi_0^{\max} > 0$.
$k \leftarrow 0$
**while** $||\nabla M(v_k)|| > 0$ **do**
    $(s^E, y^E, w^E, \mu^P, \mu^L, \mu^B) \leftarrow (s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^L, \mu_k^B)$
    Compute $v_{k+1}$ in steps 4-19 of Algorithm 5.3 until the sufficient decrease condition is achieved.
    Update $F_{\max}$.
    **if** $\chi(v_{k+1}; \mu_k^B) \leq \chi_k^{\max}$ **then** ▷ O-Iterate
        $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\frac{1}{2}\chi_k^{\max}, y_{k+1}, w_{k+1}, \mu_k^P, \mu_k^B, \tau_k)$
        $s_{k+1}^E \leftarrow \max\{0, s_{k+1}\}$
    **else if** $v_{k+1}$ satisfies (5.30 **then** ▷ M-Iterate
        $(\chi_{k+1}^{\max}, \tau_{k+1}) \leftarrow (\chi_k^{\max}, \tau_k)$
        Update $s_{k+1}^E$, $y_{k+1}^E$, $w_{k+1}^E$ using (5.31)
        **if** $\chi_{\text{feas}}(v_{k+1}) \leq \tau_k$ **then**
            $\mu_{k+1}^P \leftarrow \mu_k^P$
        **else**
            $\mu_{k+1}^P \leftarrow \frac{1}{2}\mu_k^P$
        **end if**
        **if** $\chi_{\text{comp}}(v_{k+1}; \mu_k^B) \leq \tau_k$, $s_{k+1} \geq -\tau_k e$, and $w_{k+1} \geq -\tau_k e$ **then**
            $\mu_{k+1}^B \leftarrow \mu_k^B$
        **else**
            $\mu_{k+1}^B \leftarrow \frac{1}{2}\mu_k^B$
            Reset $s_{k+1}$ and $w_{k+1}$ so that $s_{k+1} + \mu^B e > 0$ and $w_{k+1} + \mu^B e > 0$
        **end if**
    **else** ▷ F-Iterate
        $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\chi_k^{\max}, y_k^E, w_k^E, \mu_k^P, \mu_k^B, \tau_k)$
    **end if**
    Update $\mu_k^L$ via Equation (5.45).
**end while**

arbitrarily close to $-\mu^B$. In the computational setting, the logarithmic barrier terms can create errors when trying to evaluate the barrier arbitrarily close to the shifted boundary. To mitigate this issue, a so-called *fraction to the boundary rule* can be imposed. Let $\sigma$ be some parameter such that $0 < \sigma < 1$. Consider a bound $v \geq \ell$ for some variable $v$. The shifted constraint is then $v > \ell - \mu^B$. An update $v_{k+1} = v_k + d_k$ is rejected if

$$v_{k+1} < v_k - \sigma(v_k - \ell - \mu^B). \tag{5.46}$$

This condition is checked before the merit function is evaluated to prevent issues with the barrier terms in the merit function. In the implementation, $\sigma$ was chosen to be fixed throughout the algorithm, although other interior-point methods take an adaptive approach in which the fraction to the boundary parameter depends on $\mu^B$.

The final implementation detail that remains to be discussed is how to choose the initial trust-region radius $\delta_0$. Experimentally, it has been determined that this initial choice can drastically impact the performance of the algorithm, so much so that the trust-region method approach may perform worse than an inertia-controlling technique. If the initial radius is chosen to be too small, many iterations may be needed until the radius expands to a value large enough to enable rapid convergence. If the initial radius is chosen to be too large, then several iterations may be needed before the radius shrinks to a point where the model function becomes a good approximation of the merit function. For this reason, the initial trust-region radius is chosen adaptively on a problem-dependent basis by using an inertia-controlling scheme at the first iteration.

Algorithm 5.5 presents an inertia-controlling algorithm in which the matrix $H_0^M$ is modified by scalar multiples of $B_0^{(i)}$ until $H_0^M + \sigma B_0^{(i)}$ has the correct inertia.

**Algorithm 5.5.** Iterative inertia control algorithm

Given $H^M$ and $B$, $\sigma_{ic} > 0$, $\gamma_{ic} > 1$.
**if** $H^M \succ 0$ **then**
    exit
**end if**
$\sigma \leftarrow \sigma_{ic}$
**while** $H^M + \sigma B \not\succ 0$ **do**
    $\sigma \leftarrow \sigma\gamma_{ic}$
**end while**

Once a suitable value of $\sigma_0$ is found, the search direction is taken to be $d_{v,0} = (H_0^M + \sigma_0 B_0)^{-1}\nabla M(v_0)$, and the initial radius is taken to be $\delta_0 = ||d_{v,0}||_{B_0}$. The trust-region method then proceeds as presented previously. This adaptive choice ensures that a suitable initial radius is chosen. Of course, Algorithm 5.5 cannot be run as is for large-scale problems. Instead, this algorithm can replace the

Moré-Sorensen algorithm when solving the projected subproblem in each of the methods presented in Chapter 4. This simple strategy provides a reliable adaptive method for choosing the initial trust-region radius.

# Chapter 6

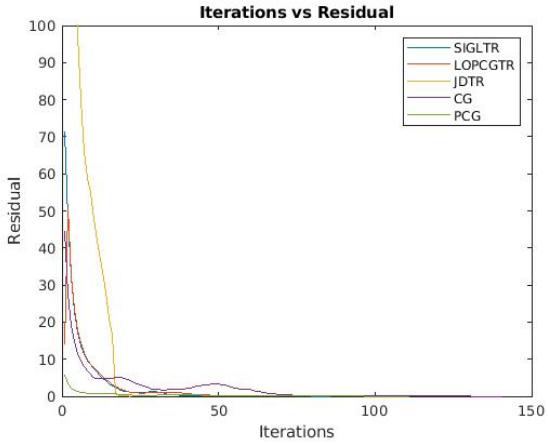
# Numerical Results

## 6.1 Comparing the Different Trust-Region Algorithms

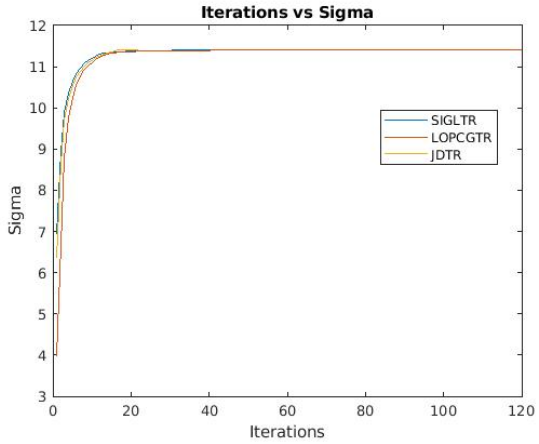
This section examines the experimental performance of the various trust-region algorithms at varying scales. All algorithms are implemented in Fortran using the 2018 standard, and are compiled with gfortran. Intel MKL is used for all BLAS and LAPACK operations. The Intel oneAPI Inspector-Executor routines are used for sparse matrix-vector multiplications. All of the experiments were performed on an Intel Core (TM) i7 CPU with 2.2GHz and 16 GM of RAM. First, consider the trust-region problem given by

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}}\ q(x) = \frac{1}{2}x^{\mathrm{T}}Hx + g^{\mathrm{T}}x \\ &\text{subject to } ||x|| \leq \delta. \end{aligned}$$

Let $\mu = ||g||/\delta - \ell$, where $\ell$ is the lower bound on the smallest eigenvalue of $H$ given by the Gershgorin circle theorem. All algorithms will choose $x_0 = 0$ as the initial estimate. Let $\tau_1 = 10^{-8}$. SIGLTR is applied with shift $\mu$. The Lanczos vectors are stored, but no reorthogonalization is applied. SIGLTR is terminated once $||r_k||_{(H+\mu I)^{-1}} \leq \tau_1||g||_{(H+\mu I)^{-1}}$. For now, SIGLTR does not use the restarting strategy. LOPCGTR uses the incomplete Cholesky decomposition of $H + \mu I$ as a preconditioner. Let $B$ denote the preconditioner. LOPCGTR is terminated once $||r_k||_B \leq \tau_1||g||_B$, and is restarted every 1000 iterations. JDTR uses the same preconditioner $B$ to solve the correction equation. The preconditioner is not updated as the value of $\sigma$ changes. JDTR is set to substitute $\sigma_j$ with $\mu$ in the correction equation for the first 15 iterations and is restarted after 50 iterations. Let $\tau_2 = 10^{-3}$. The application of preconditioned conjugate gradient to the correction equation is terminated when $||r_j||_B \leq \tau_2||r||_B$, where $r$ is the right-hand side of the correction equation. JDTR uses the same termination criteria as LOPCGTR. Additionally, conjugate-gradient is applied to the system $(H + \sigma^* I)x = -g$ with no preconditioner and with preconditioner $B$ for comparison. Recall that in exact arithmetic, GLTR converges in the same number of iterations




**(a)** Residual norm vs. iterations




**(b)** $\sigma$ vs. iterations

**Figure 6.1.** A random Gaussian trust-region subproblem.

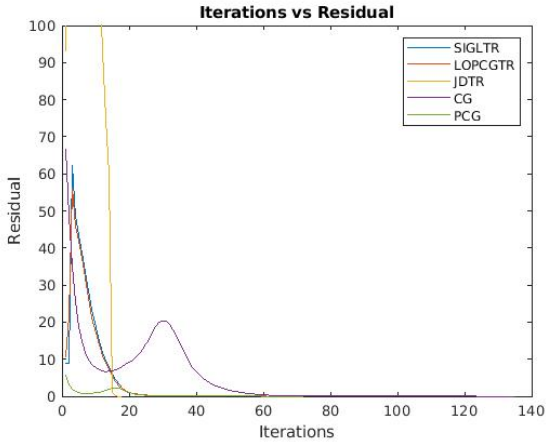
**Table 6.1.** A Gaussian random trust-region problem.

| Method | H products | M products | Preconditioner Applications | Iterations |
|---|---|---|---|---|
| SIGLTR | 0 | 87 | 86 | 86 |
| LOPCGTR | 123 | 123 | 121 | 120 |
| JDTR | 413 | 413 | 386 | 22 |
| CG | 142 | 142 | 142 | 142 |
| PCG | 112 | 112 | 112 | 112 |

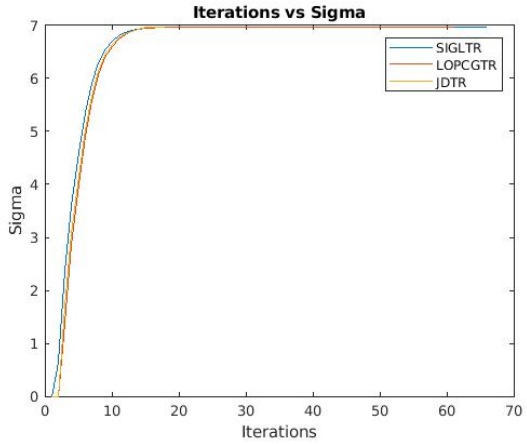
as conjugate-gradient with no preconditioner. The incomplete Cholesky decomposition is found using HSL_MI28 with all parameters set to their default values. The factorization of $H + \mu I$ used in SIGLTR is found using HSL_MA57 with all parameters set to their default values. Each solve of $H + \mu I$ uses one iterative refinement step.

For the first experiment, let $n = 10,000$, $\delta = 100$, $g$ be a Gaussian random vector with mean 0 and variance 1, and $H$ be a random sparse matrix with an average of 25 nonzero entries per column, whose entries are Gaussian random with mean 0 and variance 1. This is a relatively well-conditioned problem. For this reason, it is expected that standard conjugate-gradient, and therefore GLTR, shall perform reasonably well and that the number of iterations is not dramatically improved by the new methods. See Figure 6.1 and Table 6.1.

For the second experiment, let all parameters remain the same except for the objective matrix $H$. Let $H = \widehat{H} + D$, where $\widehat{H}$ is the same random matrix as before, and $D$ is a diagonal matrix such that $(D)_{i,i} = i^{3/2}/n - 1$. This additional diagonal term makes the problem slightly more ill-conditioned. Thus, it is expected that the new algorithms will show more noticeable improvements over GLTR. See Figure




**(a)** Residual norm vs. iterations




**(b)** $\sigma$ vs. iterations

**Figure 6.2.** A random Gaussian trust-region subproblem with a small diagonal offset.

**Table 6.2.** A Gaussian random trust-region problem with a small diagonal offset.

| Method | H products | M products | Preconditioner Applications | Iterations |
|---|---|---|---|---|
| SIGLTR | 0 | 69 | 67 | 66 |
| LOPCGTR | 64 | 64 | 64 | 62 |
| JDTR | 155 | 155 | 134 | 18 |
| CG | 136 | 136 | 136 | 136 |
| PCG | 73 | 73 | 73 | 73 |

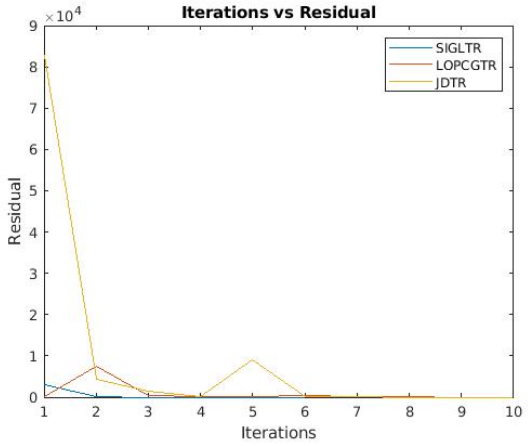
6.2 and Table 6.2.

For the third experiment, let all parameters remain the same, except for the diagonal offset to the objective matrix $H$, which is now set to $(D)_{i,i} = i^2 - n$. This matrix is extremely ill-conditioned and is dominated by the $D$ term. The extreme ill-conditioning of this system all but guarantees that conjugate-gradient fails to converge. It is worth noting that the eigenvalues of the matrix $H$ are clustered toward the low end of the spectrum, which may help the convergence of all methods considered. For the instance of this problem shown, $\sigma \approx 9998.9$, and $|\sigma - \mu| \approx 35.7$. Thus, $\mu$ is relatively close to $\sigma$. See Table 6.3 for details. The plots of the residual norms and Lagrange multiplier values are not particularly illustrative for this problem and are excluded.

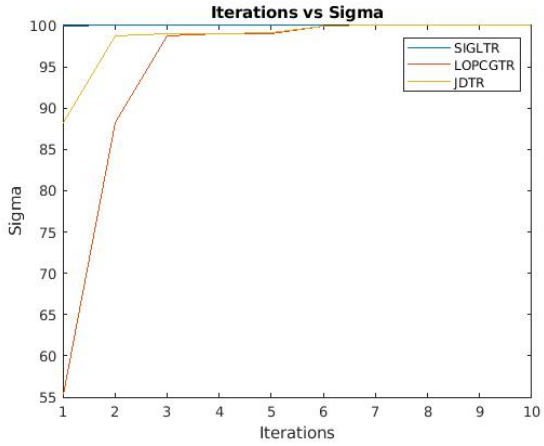
For the next experiment, let $\delta = 1,000$, and $H$ be the diagonal matrix such that $(H)_{i,i} = i - 101$. Let $g$ be a Gaussian random vector with mean 0 and variance 1 such that $[g]_1 = 0$, thus making this problem an instance of the hard case. The optimal value of $\sigma$ is then 100. Conjugate gradient and preconditioned conjugate gradient are not used in this comparison, as the matrix $H + \sigma^* I$ is singular. SIGLTR is run with a block size of 2 to guarantee convergence. See Figure 6.3 and Table 6.4. Each

**Table 6.3.** A Gaussian random trust-region problem with a large diagonal offset.

| Method | H products | M products | Preconditioner Applications | Iterations |
|---|---|---|---|---|
| SIGLTR | 0 | 21 | 18 | 18 |
| LOPCGTR | 20 | 20 | 18 | 17 |
| JDTR | 72 | 72 | 72 | 16 |
| CG | +10,000 | +10,000 | +10,000 | +10,000 |
| PCG | 21 | 21 | 21 | 21 |




**(a)** Residual norm vs. iterations




**(b)** $\sigma$ vs. iterations

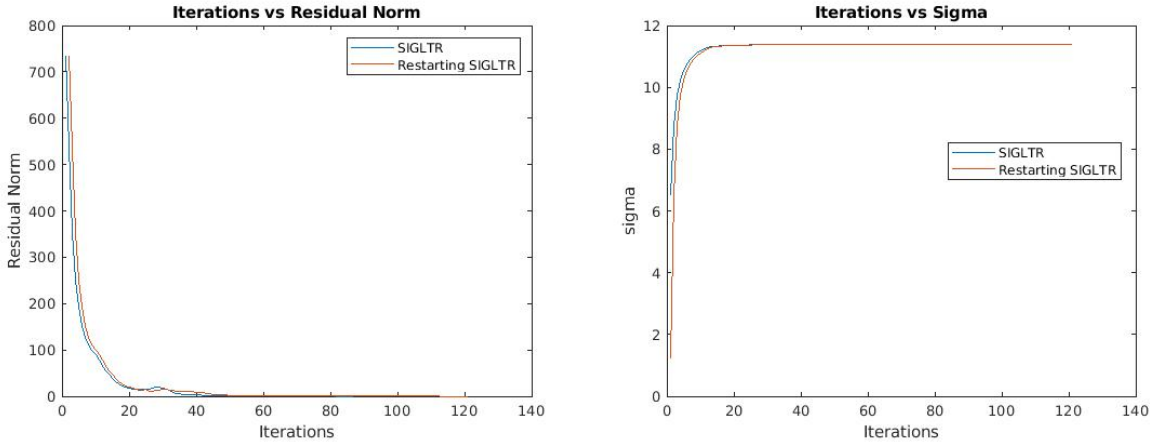
**Figure 6.3.** Hard Case.

algorithm was able to converge to the actual solution, and $\sigma$ was computed to be equal to the left-most eigenvalue.

These experiments reveal that the best choice for an arbitrary trust-region problem using the identity matrix to define the trust region is LOPCGTR. Its ability to use the incomplete Cholesky decomposition of $H + \mu I$ as a preconditioner gives the method a distinct advantage over GLTR. Conversely, the fact that LOPCGTR does not require a full factorization of $H + \mu I$ gives a distinct advantage over SIGLTR. Although SIGLTR has the benefit of not requiring any multiplications of $H$, this is offset by the fact that linear systems of the form $(H + \mu I)u = v$ must be solved at each iteration. This operation is significantly more expensive than applying the incomplete Cholesky decomposition. Moreover, if the

**Table 6.4.** Hard Case.

| Method | H products | M products | Preconditioner Applications | Iterations |
|---|---|---|---|---|
| SIGLTR | 0 | 17 | 14 | 6 |
| LOPCGTR | 13 | 13 | 11 | 10 |
| JDTR | 43 | 43 | 30 | 10 |




**(a)** Residual norm vs. iterations

**(b)** $\sigma$ vs. iterations

**Figure 6.4.** SIGLTR vs Restarting SIGLTR.

block size is taken to be 2 in SIGLTR, even more matrix-vector operations are required, although this is offset by the fact that most implementations of these libraries heavily optimize the application to vector blocks. Although JDTR tends to converge in fewer iterations, each iteration requires more applications of $H$ and the precondition, which usually means more work for JDTR.

For the next experiment, the restarting of SIGLTR is examined. Let $g$ be a Gaussian random vector with mean 0 and variance 1, $M = I$, $\delta = 1000$, and $H$ a matrix with on average 25 nonzero entries per column, whose values are Gaussian random variables with mean 0 and variance 1. First, SIGLTR is run with a block size of 2 and no restarting. Next, it is set to restart every 25 iterations. This is a very aggressive restart strategy. In practice, it is advisable to restart far less often. See Figure 6.4. These graphs indicate that restarting, even frequently, only negligibly hinders convergence.

For the next experiment, consider the doubly-augmented problem

$$\begin{aligned}
&\min_{d_x \in \mathbb{R}^n, d_y \in \mathbb{R}^m} \begin{pmatrix} g_x^{\mathrm{T}} & g_y^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \frac{1}{2} \begin{pmatrix} x^{\mathrm{T}} & y^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} H + 2J^{\mathrm{T}} D^{-1} J & -J^{\mathrm{T}} \\ -J & D \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \\
&\text{subject to} \quad \begin{pmatrix} x^{\mathrm{T}} & y^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} I + 2J^{\mathrm{T}} D^{-1} J & -J^{\mathrm{T}} \\ -J & D \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \delta^2,
\end{aligned}$$

where $g = (g_x, g_y)$ is a Gaussian random vector with mean 0 and variance 1, $\delta = 1,000$, $J \in \mathbb{R}^{m \times n}$ is a sparse Gaussian random matrix with an average of 5 nonzero entries per column with mean 0 and variance 1, $H \in \mathbb{R}^{n \times n}$ is a sparse Gaussian random matrix with an average of 5 nonzero entries per column with mean 0 and variance 1 plus a diagonal matrix $E$ given by $(E)_{i,i} = i^{3/2} - 100$, and $D \in \mathbb{R}^{m \times m}$ is a random

**Table 6.5.** A Doubly-Augmented Trust-Region Problem.

| Method | H products | J products | Preconditioner Applications | Iterations |
|---|---|---|---|---|
| SIGLTR | 0 | 398 | 78 | 77 |
| LOPCGTR | 146 | 582 | 144 | 143 |
| JDTR | 483 | 1091 | 440 | 41 |

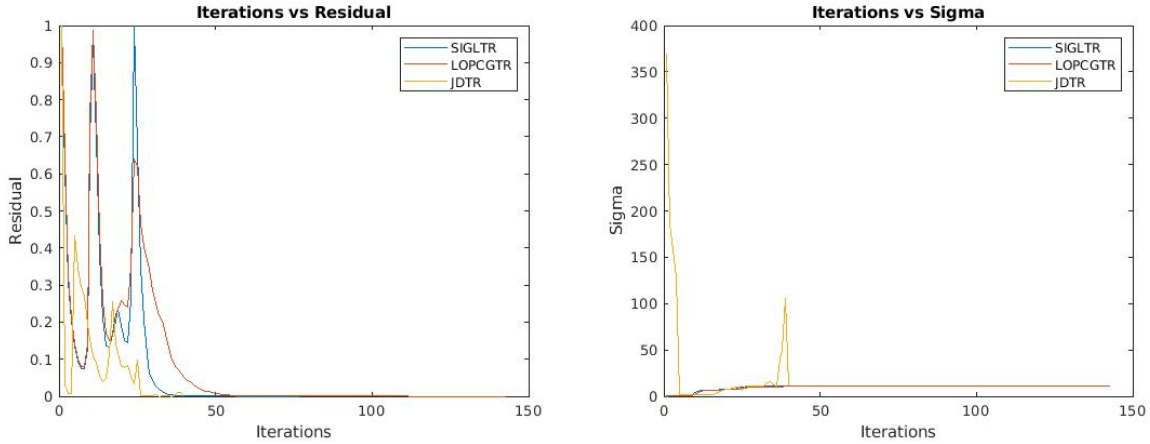
diagonal matrix whose entries are the absolute value Gaussian random variables with mean $10^{-3}$ and variance 0. Let $n = 10,000$, and $m = 9,999$. Let $H^M$ denote the objective matrix, and $M$ denote the constraint matrix. Let $\mu = g^{\mathrm{T}}g/(\delta||g||_M) - \lambda_n(H)$, where $\lambda_n(H)$ is the leftmost eigenvalue of $H$. As before, SIGLTR is used with shift $\mu$, stored Lanczos vectors, and no reorthogonalization. LOPCGTR is run with the matrix

$$B = \begin{pmatrix} \mathrm{diag}(H + \mu I) + (1+\mu)2J^{\mathrm{T}}D^{-1}J & -(1+\mu)J^{\mathrm{T}} \\ -(1+\mu)J & (1+\mu)D \end{pmatrix}$$

implicitly used as a preconditioner. JDQZTR is applied using the signed incomplete Cholesky decomposition of the fixed matrix

$$C = \begin{pmatrix} H + \mu I & (1+\mu)J^{\mathrm{T}} \\ (1+\mu)J & -(1+\mu)D \end{pmatrix}$$

as a preconditioner for the correction equation. The signed incomplete Cholesky decomposition is found with HSL_MI30 with all parameters set to default values. The correction equation is solved using restarted GMRES with a relative convergence tolerance of $\tau_2 = 10^{-3}$. Due to the fact that JDQZTR measures the residual in the Euclidean norm, the results in 6.5 are normalized by dividing by the largest value that the residual norm takes as the algorithm proceeds. This example shows that for both SIGLTR and LOPCGTR, it is possible for the residual to increase in between iterations. However, once the correct value of $\sigma$ is found, the residual seems to be strictly decreasing. This example also shows that the value of $\sigma$ does not strictly increase in the JDQZTR algorithm, as the Petrov-Galerkin condition does not preserve the Cauchy interlacing property. The results of this experiment are more challenging to interpret. Both SIGLTR and LOPCGTR require one call to HSL_MA57. This constitutes the majority of the execution time for this problem. LOPCGTR required the factorization of a less dense matrix. However, it required almost twice as many iterations. JDTR required the fewest iterations but significantly more matrix-vector products with $H$ and $J$. Overall, it seems that the best choice of algorithm is problem dependent. That being said, the instability of the JDQZTR algorithm suggests that it should only be used in problems so




**(a)** Residual norm vs. iterations

**(b)** $\sigma$ vs. iterations

**Figure 6.5.** A Doubly-Augmented Trust-Region Problem.

large that a single call to HSL_MA57 is prohibitively expensive. For instance, consider the same problem as above, but with $\delta = 10$, $n = 100,000$, and $m = 99,999$. JDQZTR was able to converge for this problem in 29 iterations and required 264 multiplications with $H$, 618 multiplications with $J$, and 232 applications of the preconditioner.

On the other hand, both SIGLTR and LOPCGTR could not complete the single factorization of their respective preconditioners without exhausting the available memory of the workstation. At the same time, it is well known that GMRES can converge very slowly when applied to saddle-point systems, even when preconditioning is used. For these reasons, in most cases, SIGLTR and LOPCGTR are recommended over GLTR and JDQZTR.

## 6.2 The Shifted Primal-Dual Interior-Point Algorithm

This section examines the experimental performance of the all-shifted primal-dual penalty-barrier trust-region method. The algorithm is run on the CUTEst NLP problem collection, consisting of problems with nonlinear objective functions, bound constraints, linear constraints, and nonlinear constraints, ranging from low dimension (less than five variables and constraints) to large dimension (greater than 50,000 variables and constraints). Analysis of the results is carried out with performance profiles, see [9]. See Table 6.6 for the parameters used in the implementation of Algorithm 5.1 and Algorithm 5.2.

The first test examines the performance between using the diagonal trust-region matrix $M^{(1)}$ vs $M^{(2)}$. Intuitively, it stands to reason that $M^{(2)}$ will yield better results, as the trust-region subproblems are better conditioned, and as the trust-region radius converges to zero, the search direction still takes into

**Table 6.6.** Algorithm 5.3 and 5.4 Parameters

| Parameter | Description | Value |
|---|---|---|
| $\eta_1$ | Successful trust-region iteration parameter | 0.001 |
| $\eta_2$ | Very successful trust-region iteration parameter | 0.25 |
| $\eta_{F,1}$ | Successful trust-region iteration parameter for decrease in path-following equations | 0.9 |
| $\eta_{F,2}$ | Very successful trust-region iteration parameter for decrease in path-following equations | 0.45 |
| $\gamma_C$ | Trust region contraction parameter | 0.5 |
| $\gamma_E$ | Trust region expansion parameter | 2.0 |
| $s_{\max}$ | Safeguard for $s^E$ | $1.0 \times 10^6$ |
| $y_{\max}$ | Safeguard for $y^E$ | $1.0 \times 10^6$ |
| $w_{\max}$ | Safeguard for $w^E$ | $1.0 \times 10^6$ |
| $\mu_0^P$ | Initial penalty parameter | 0.001 |
| $\mu_0^L$ | Initial flexible trust-region parameter | 0.1 |
| $\mu_0^B$ | Initial barrier parameter | 0.001 |
| $\sigma$ | Fraction to the boundary parameter | 0.9 |
| $\sigma_{ic}$ | Initial inertia control modification | 0.1 |
| $\gamma_{ic}$ | Inertia control expansion parameter | 1.5 |
| $f_{\text{unbounded}}$ | Unbounded objective function tolerance | $-1.0 \times 10^9$ |
| $M_{\max}$ | Constant in sufficient decrease tolerance | $1.0 \times 10^{15}$ |
| $F_{\max}$ | Initial constant in sufficient decrease tolerance | $1.0 \times 10^8$ |
| $\chi_0^{\max}$ | Initial O-iterate tolerance | 1000 |
| $\tau_0$ | Initial M-iterate tolerance | 0.1 |
| $\tau_P$ | Primal feasibility tolerance | $1.0 \times 10^{-5}$ |
| $\tau_D$ | Dual feasibility tolerance | $1.0 \times 10^{-5}$ |
| $\tau_{\text{inf}}$ | Infeasible stationary-point tolerance | $1.0 \times 10^{-5}$ |

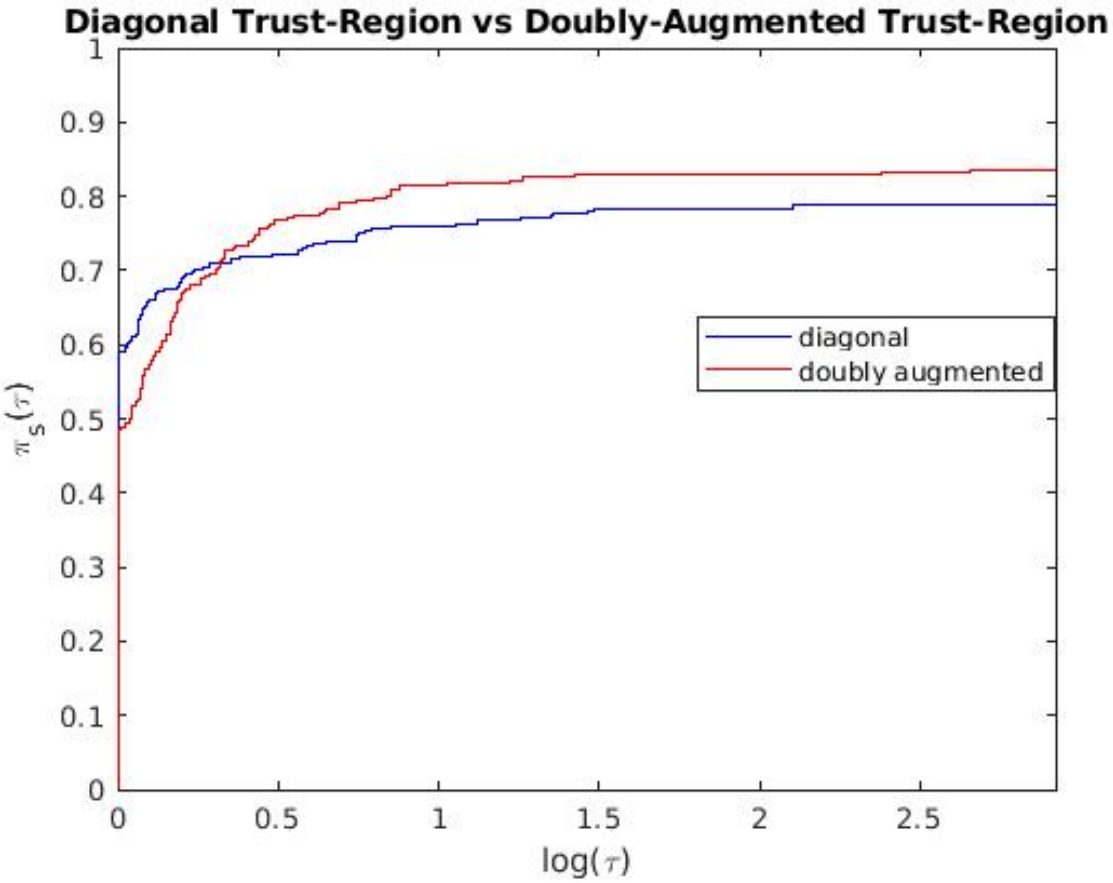



**Figure 6.6.** Performance Profile of function evaluations used in all-shifted primal-dual penalty-barrier trust-region method with different trust-region matrices

account some of the curvature information of the constraints. On the other hand, when using $M^{(1)}$, as the trust-region radius decreases, the search directions converge to the gradient-descent direction, which is well known to yield poor search directions in interior-point methods. This experiment is run with the Moré-Sorensen method for solving the trust-region subproblem. The method was considered to have failed if it required more than 500 iterations or if the execution took longer than 30 minutes. The performance profiles shown here are of the number of evaluations of the objective function and objective function gradient. The results are essentially identical when the performance profiles are of the number of inner iterations. See Figure 6.6. Disappointingly, the result is not uniform, as some problems converged in fewer iterations with the diagonal trust-region matrix. However, the results indicate that the doubly-augmented trust-region matrix is the preferred choice.

Table 6.7 shows the number of iterations and function evaluations of the all-shifted primal-dual penalty-barrier trust-region method using SIGLTR with the doubly-augmented trust-region matrix to

solve the trust-region subproblem on the Hock-Schittkowski [20] problems. These simpler problems tend to yield a good initial benchmark of how well an algorithm will perform before applying it to larger, more complicated problems. The iterations column only reflects the number of successful trust-region iterations. The $f_{\text{evals}}$ column counts the number of objective function evaluations. This count coincides with the number of total trust-region iterations, as one function evaluation occurs per trust-region iteration. Simple bounds on the primal variables are not included in the number of constraints.

The next experiment compares the performance of the all-shifted primal-dual penalty-barrier trust-region method with two different implementations to guarantee convergence of the inner iterations. The first is a hybrid trust-region line-search method. This method follows the method presented in [12]. The hybrid trust-region line-search method differs from a pure trust-region method in that progress is made even in unsuccessful iterations by performing a backtracking line search or a Wolfe line search on the search direction computed by solving the trust-region subproblem if the iteration is unsuccessful. The idea behind this method is that fewer trust-region subproblems need to be solved in such a method than in a pure trust-region approach at the expense of more iterations. The second method is similar to the pure trust-region method, however after an unsuccessful iteration yields a search direction $d_{v,k}$, subsequent trust-region subproblems are projected onto the subspace $\text{span}\{d_{v,k}, (B^{(2)})^{-1}\nabla M\}$. This two-dimensional subproblem is trivial to solve. Therefore this method does not require significantly more computation than the hybrid trust-region line-search method. This experiment used the doubly-augmented trust-region matrix and the Moré-Sorensen algorithm. See Figure 6.7. As expected, the trust-region method performs best overall. However, the additional computation required by the Moré-Sorenson algorithm makes this method prohibitively expensive. The SIGLTR and LOPCGTR methods enormously help to alleviate this overhead, particularly when their warm-starting capabilities are taken advantage of. Interestingly, the JDQZTR algorithm tended to perform slower than even the Moré-Sorenson algorithm when applied to these real problems. This is perhaps because GMRES tends to converge quite slowly on saddle-point problems, and thus even if JDQZTR requires fewer iterations than other methods, the cost of each iteration outweighs this benefit. For this reason, JDQZTR is not compared here. Instead, JDQZTR should be considered a last resort for problems that are too large to handle by other methods.

First, the number of function evaluations between the method using the Moré-Sorensen algorithm and the SIGLTR algorithm are compared in Figure 6.8. Interestingly, using the SIGLTR algorithm led to better overall performance, even though the same trust-region subproblems are being solved. This could be because SIGLTR yields more accurate trust-region solutions. More likely, the faster execution time of SIGLTR meant that fewer problems were deemed to have failed.

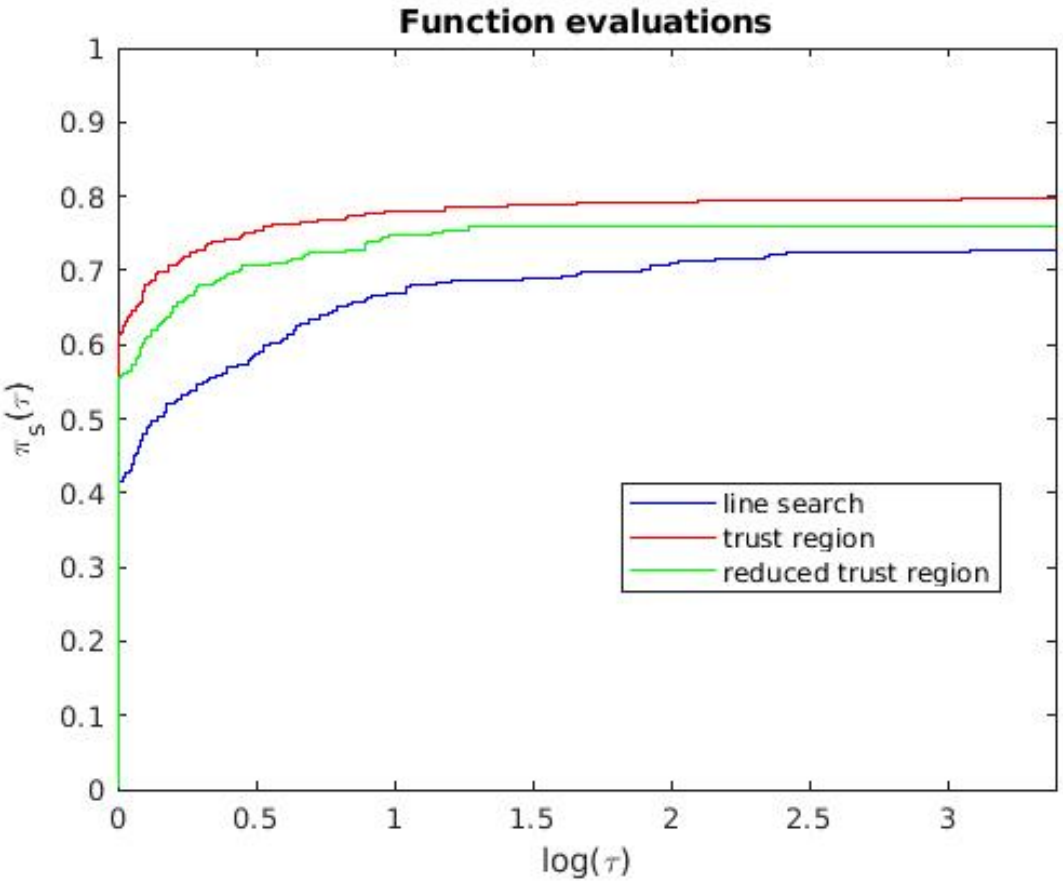



**Figure 6.7.** Performance Profile of function evaluations used in all-shifted primal-dual penalty-barrier trust-region method with different inner iterations strategies

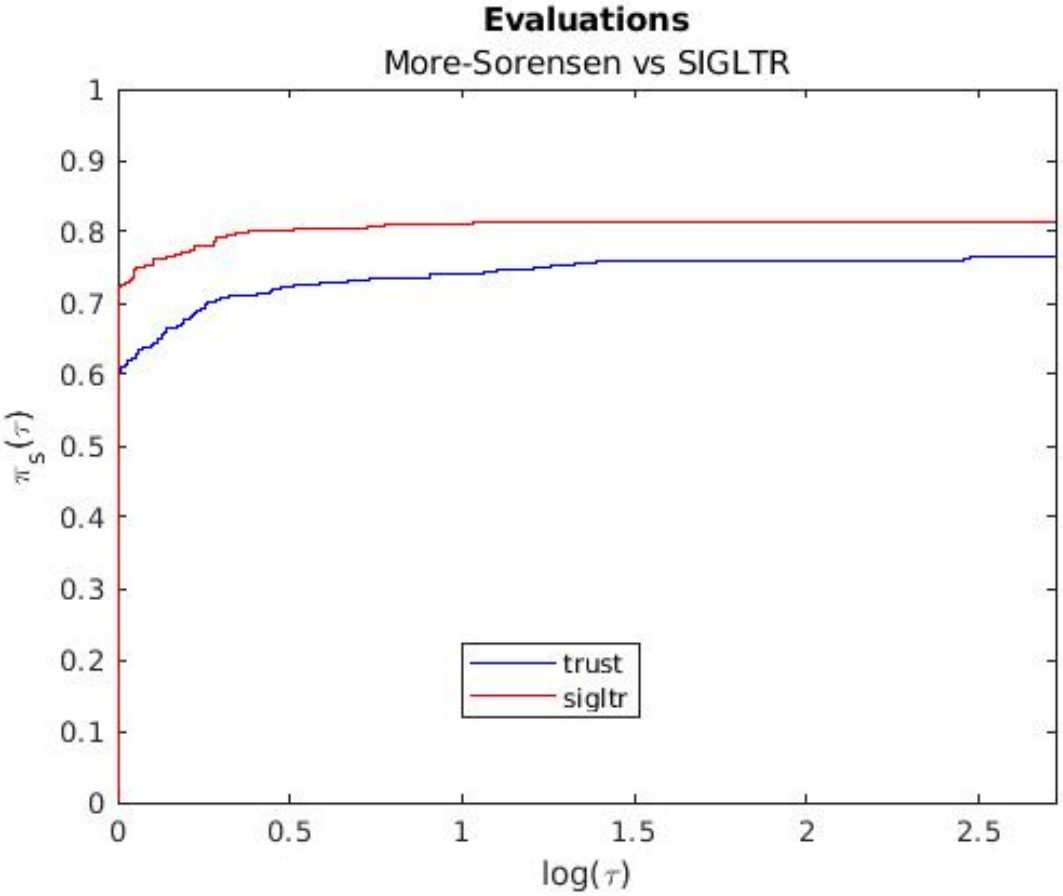



**Figure 6.8.** Performance Profile of function evaluations used in all-shifted primal-dual penalty-barrier trust-region method with Moré Sorensen vs. SIGLTR

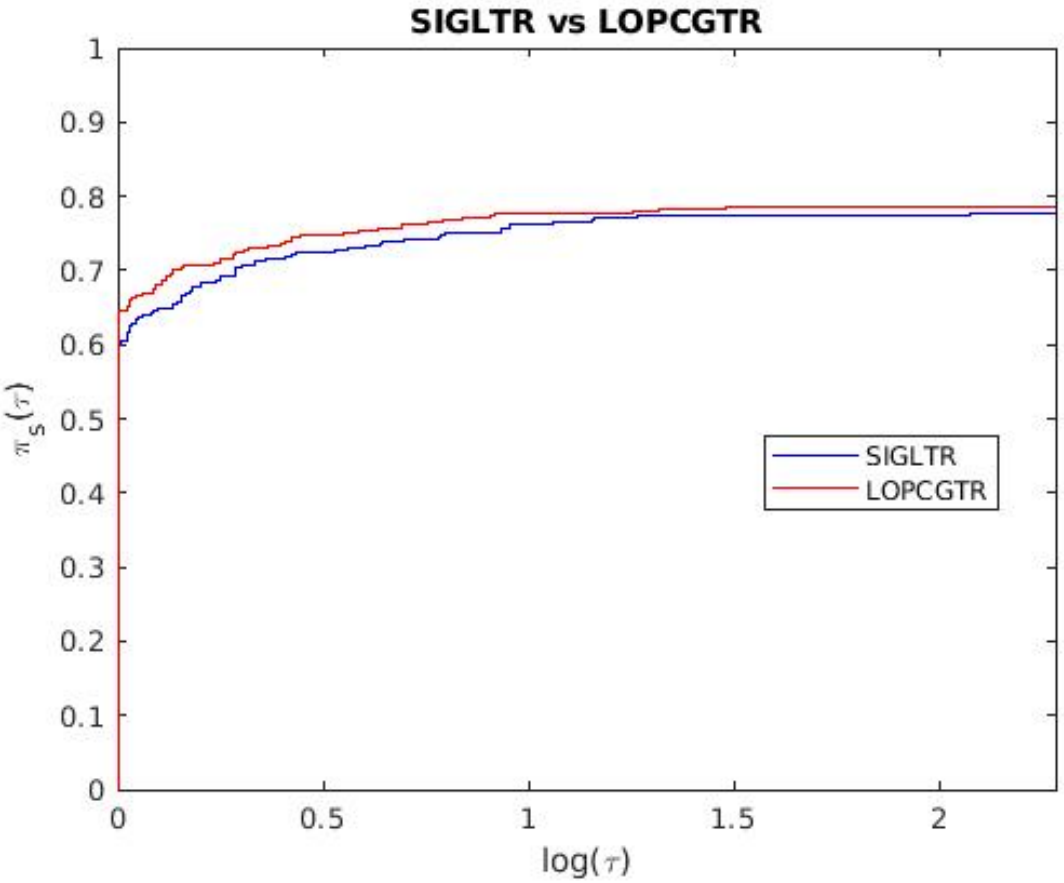



**Figure 6.9.** Performance Profile of function evaluations used in all-shifted primal-dual penalty-barrier trust-region method with SIGLTR vs. LOPCGTR

**Table 6.7.** Hock-Schittkowski Results

| Problem | Variables | Constraints | Iterations | $f_{\text{evals}}$ |
|---|---|---|---|---|
| HS6 | 2 | 1 | 20 | 36 |
| HS7 | 2 | 1 | 9 | 13 |
| HS8 | 2 | 2 | 4 | 6 |
| HS9 | 2 | 1 | 4 | 5 |
| HS10 | 2 | 1 | 9 | 10 |
| HS11 | 2 | 1 | 8 | 9 |
| HS12 | 2 | 1 | 28 | 40 |
| HS13 | 2 | 1 | 12 | 13 |
| HS14 | 2 | 2 | 8 | 9 |
| HS15 | 2 | 2 | 71 | 78 |
| HS16 | 2 | 2 | 8 | 9 |
| HS17 | 2 | 2 | 9 | 10 |
| HS18 | 2 | 2 | 22 | 35 |
| HS19 | 2 | 2 | 29 | 30 |
| HS20 | 2 | 3 | 18 | 19 |
| HS21 | 2 | 1 | 8 | 9 |
| HS21MOD | 7 | 1 | 11 | 12 |
| HS22 | 2 | 2 | 8 | 9 |
| HS23 | 2 | 5 | 15 | 16 |
| HS24 | 2 | 3 | 9 | 10 |
| HS26 | 3 | 1 | 25 | 26 |
| HS27 | 3 | 1 | 47 | 59 |
| HS28 | 3 | 1 | 1 | 2 |
| HS29 | 3 | 1 | 65 | 114 |
| HS30 | 3 | 1 | 7 | 8 |
| HS31 | 3 | 1 | 8 | 9 |
| HS32 | 3 | 2 | 11 | 12 |
| HS33 | 3 | 2 | 51 | 97 |
| HS34 | 3 | 2 | 10 | 11 |

**Table 6.7.** (cont.)

| | | | | |
|---|---|---|---|---|
| HS35 | 3 | 1 | 7 | 8 |
| HS35I | 3 | 1 | 7 | 8 |
| HS35MOD | 3 | 1 | 9 | 10 |
| HS36 | 3 | 1 | 12 | 13 |
| HS37 | 3 | 2 | 12 | 13 |
| HS39 | 4 | 2 | 8 | 9 |
| HS40 | 4 | 3 | 3 | 4 |
| HS41 | 4 | 1 | 8 | 9 |
| HS42 | 4 | 2 | 4 | 5 |
| HS43 | 4 | 3 | 19 | 20 |
| HS44 | 4 | 6 | 9 | 10 |
| HS44NEW | 4 | 6 | 9 | 10 |
| HS46 | 5 | 2 | 19 | 20 |
| HS47 | 5 | 3 | 29 | 44 |
| HS48 | 5 | 2 | 1 | 2 |
| HS49 | 5 | 2 | 13 | 14 |
| HS50 | 5 | 3 | 8 | 9 |
| HS51 | 5 | 3 | 1 | 2 |
| HS52 | 5 | 3 | 2 | 3 |
| HS53 | 5 | 3 | 6 | 7 |
| HS54 | 6 | 1 | 6 | 7 |
| HS55 | 6 | 6 | 6 | 7 |
| HS56 | 7 | 4 | 13 | 20 |
| HS57 | 2 | 1 | 105 | 153 |
| HS59 | 2 | 3 | 14 | 17 |
| HS60 | 3 | 1 | 6 | 7 |
| HS61 | 3 | 2 | 14 | 22 |
| HS62 | 3 | 1 | 18 | 19 |
| HS63 | 3 | 2 | 11 | 16 |
| HS64 | 3 | 1 | 24 | 25 |

**Table 6.7.** (cont.)

| | | | | |
|---|---|---|---|---|
| HS65 | 3 | 1 | 47 | 68 |
| HS66 | 3 | 2 | 7 | 8 |
| HS67 | 3 | 14 | 17 | 18 |
| HS68 | 4 | 2 | 20 | 22 |
| HS69 | 4 | 2 | 11 | 12 |
| HS70 | 4 | 1 | 14 | 21 |
| HS71 | 4 | 2 | 12 | 13 |
| HS72 | 4 | 2 | 26 | 27 |
| HS73 | 4 | 3 | 14 | 15 |
| HS74 | 4 | 5 | 22 | 13 |
| HS75 | 4 | 5 | 33 | 34 |
| HS76 | 4 | 3 | 9 | 10 |
| HS76I | 4 | 3 | 9 | 10 |
| HS77 | 5 | 2 | 35 | 47 |
| HS78 | 5 | 3 | 5 | 6 |
| HS79 | 5 | 3 | 6 | 7 |
| HS80 | 5 | 3 | 6 | 8 |
| HS81 | 5 | 3 | 7 | 8 |
| HS83 | 5 | 3 | 27 | 28 |
| HS84 | 5 | 3 | 26 | 27 |
| HS85 | 5 | 21 | 3 | 4 |
| HS86 | 5 | 10 | 13 | 14 |
| HS87 | 6 | 4 | 74 | 96 |
| HS88 | 2 | 1 | 62 | 84 |
| HS89 | 3 | 1 | 62 | 85 |
| HS90 | 4 | 1 | 62 | 97 |
| HS91 | 5 | 1 | 53 | 82 |
| HS92 | 6 | 1 | 55 | 88 |
| HS93 | 6 | 2 | 16 | 17 |
| HS95 | 6 | 4 | 18 | 20 |

**Table 6.7.** (cont.)

| | | | | |
|---|---|---|---|---|
| HS96 | 6 | 4 | 49 | 79 |
| HS97 | 6 | 4 | 15 | 21 |
| HS98 | 6 | 4 | 16 | 24 |
| HS99 | 7 | 2 | 13 | 14 |
| HS99EXP | 31 | 21 | 28 | 32 |
| HS100 | 7 | 4 | 28 | 37 |
| HS100LNP | 7 | 2 | 16 | 21 |
| HS100MOD | 7 | 4 | 10 | 13 |
| HS101 | 7 | 5 | 51 | 77 |
| HS102 | 7 | 5 | 71 | 104 |
| HS103 | 7 | 5 | 35 | 48 |
| HS104 | 8 | 5 | 9 | 10 |
| HS105 | 8 | 1 | 19 | 20 |
| HS106 | 8 | 6 | 64 | 71 |
| HS107 | 9 | 6 | 31 | 34 |
| HS108 | 9 | 13 | 28 | 54 |
| HS109 | 9 | 10 | 39 | 41 |
| HS111 | 10 | 3 | 15 | 19 |
| HS111LNP | 10 | 3 | 32 | 52 |
| HS112 | 10 | 3 | 7 | 8 |
| HS113 | 10 | 8 | 133 | 222 |
| HS114 | 10 | 11 | 21 | 22 |
| HS116 | 13 | 14 | 99 | 104 |
| HS117 | 15 | 5 | 16 | 17 |
| HS118 | 15 | 17 | 21 | 22 |
| HS119 | 16 | 8 | 17 | 18 |
| HS268 | 5 | 5 | 15 | 15 |

Finally, the number of function evaluations required by the outer algorithm when using SIGLTR and LOPCGTR are compared in Figure 6.9. This plot shows that the two methods yield comparable results.

In conclusion, the methods presented in this dissertation yield noticeable improvements over existing trust-region methods when applied to interior-point methods. In particular, the SIGLTR and LOPCGTR methods exhibit exceptional performance in all cases, especially compared to existing methods. Both methods have strong theoretical guarantees and can be restarted and warm-started to accelerate convergence and improve stability. These two methods make pure trust-region implementations of interior-point methods significantly more practical than traditional methods, allowing interior-point methods to take advantage of the strong theoretical results of trust-region methods. The JDTR algorithm, while showing improvements over existing methods, does not exhibit the performance of the SIGLTR or LOPCGTR methods. For unconstrained problems, trust-region methods using SIGLTR and LOPCGTR should be attempted before JDTR. JDQZTR should only be used for constrained problems if SIGLTR and LOPCGTR cannot be applied.

# Bibliography


[1] Satoru Adachi, Satoru Iwata, Yuji Nakatsukasa, and Akiko Takeda. "Solving the Trust-Region Subproblem By a Generalized Eigenvalue Problem". *SIAM Journal on Optimization* 27.1 (2017), pp. 269–291.

[2] Roberto Andreani, José Mario Martínez, and B. F. Svaiter. "A new sequential optimality condition for constrained optimization and algorithmic consequences". *SIAM J. Optim.* 20.6 (2010), pp. 3533–3554.

[3] Richard Beals and Roderick Wong. *Special Functions and Orthogonal Polynomials*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2016.

[4] I. Bongartz, A. R. Conn, N. I. M. Gould, and Philippe L. Toint. "CUTE: Constrained and unconstrained testing environment". *ACM Trans. Math. Software* 21.1 (1995), pp. 123–160.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[6] James R. Bunch and Linda Kaufman. "Some stable methods for calculating inertia and solving symmetric linear systems". *Mathematics of Computation* 31 (1975), pp. 163–179.

[7] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-Region Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2000, pp. xx+959.

[8] Gerard Debreu. "Definite and semidefinite quadratic forms". *Econometrica* 20 (1952), pp. 295–300.

[9] Elizabeth D. Dolan and Jorge J. Moré. "Benchmarking optimization software with performance profiles". *Math. Program.* 91.2, Ser. A (2002), pp. 201–213.

[10] Diederik R. Fokkema, Gerard L. G. Sleijpen, and Henk A. Van der Vorst. "Jacobi–Davidson Style QR and QZ Algorithms for the Reduction of Matrix Pencils". *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 94–125.

[11] Anders Forsgren and Philip E. Gill. "Primal-Dual Interior Methods for Nonconvex Nonlinear Programming". *SIAM Journal on Optimization* 8.4 (1998), pp. 1132–1152.

[12] E. Michael Gertz and Philip E. Gill. *A primal-dual trust-region algorithm for nonlinear programming*. Numerical Analysis Report NA 02-1. University of California, San Diego, 2002.

[13] Philip E. Gill, Vyacheslav Kungurtsev, and Daniel P. Robinson. *A Shifted Primal-Dual Penalty-Barrier Method for Nonlinear Optimization*. Center for Computational Mathematics Report CCoM 19-03. University of California, San Diego, 2019.

[14] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Numerical Linear Algebra and Optimization, volume 1*. Redwood City: Addison-Wesley Publishing Company, 1991.

[15] Philip E. Gill and Daniel P. Robinson. "A primal-dual augmented Lagrangian". *Computational Optimization and Applications* 51.1 (Jan. 2012), pp. 1–25.

[16] Philip E. Gill and Margaret H. Wright. *Computational Optimization: Nonlinear Programming*. To be published in 2023. New York, NY, USA: Cambridge University Press, 2023.

[17] Philip E. Gill and Minxin Zhang. *A Projected-Search Interior Method for Nonlinear Optimization*. Center for Computational Mathematics Report CCoM 22-01. La Jolla, CA: Center for Computational Mathematics, University of California, San Diego, 2022.

[18] Nicholas I. M. Gould, Stefano Lucidi, Massimo Roma, and Philippe L. Toint. "Solving the Trust-Region Subproblem using the Lanczos Method". *SIAM Journal on Optimization* 9.2 (1999), pp. 504–525.

[19] Nicholas I. M. Gould, Dominique Orban, and Philippe L. Toint. "CUTEr and SifDec: A constrained and unconstrained testing environment, revisited". *ACM Trans. Math. Softw.* 29 (2003), pp. 373–394.

[20] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*. Lecture Notes in Econom. Math. Syst. 187. Berlin: Springer-Verlag, 1981.

[21] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[22] Zhongxiao Jia. "The Convergence of Generalized Lanczos Methods for Large Unsymmetric Eigenproblems". *SIAM Journal on Matrix Analysis and Applications* 16.3 (1995), pp. 843–862.

[23] Zhongxiao Jia and Fa Wang. *The convergence of the Generalized Lanczos Trust-Region Method for the Trust-Region Subproblem*. 2019.

[24] Chih-Jen Lin and Jorge J. Moré. "Incomplete Cholesky factorizations with limited memory". *SIAM J. Sci. Comput.* 21.1 (1999), pp. 24–45.

[25] Jorge J. Moré and Danny C. Sorensen. "Computing a trust region step". *SIAM J. Sci. and Statist. Comput.* 4 (1983), pp. 553–572.

[26] C. C. Paige. "The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices". PhD thesis. University of London, 1971.

[27] Horst Simon. "The Lanczos Algorithm With Partial Reorthogonalization". *Mathematics of Computation - Math. Comput.* 42 (Jan. 1984), pp. 115–115.