UNIVERSITY OF CALIFORNIA SAN DIEGO

Cubic Regularization Algorithms for Unconstrained and Constrained Optimization

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

in

Mathematics

by

Ziyan Zhu

2023

DEDICATION

xxx

TABLE OF CONTENTS

<div align="center">LIST OF FIGURES</div>

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my PhD advisor, Philip E. Gill, for his invaluable guidance, patience, and support throughout my research. His expertise and encouragement were instrumental in shaping this thesis, and I am deeply grateful for the opportunities he provided me to attend conferences and talks, which expanded my knowledge and research skills.

I would also like to thank my committee members, Randolph E. Bank, Li-Tien Cheng, Hyunsun Alicia Kim, and Danna Zhang, for their time, expertise, and support. I am especially grateful for their flexibility and accommodation with scheduling the defense, which allowed me to successfully defend my thesis.

I would like to extend my thanks to Elizabeth Wong for her computer expertise and assistance in setting up my laptop for the testing environments. I am also grateful for her invitation to give a talk at the INFORMS conference, an experience that has enriched my academic career.

On a personal note, my heartfelt thanks go to my parents, whose unwavering love and support have been the foundation of my achievements. I could not have come this far without their constant encouragement and belief in my abilities.

My furry companion, PP, deserves a special mention for her endless emotional support, especially during the difficult period of the COVID-19 pandemic. Her presence by my side made me feel less alone, and has been a constant source of comfort and joy.

Last but not least, I would like to thank my girlfriend, Xinyi Li, for her love, support, and constant encouragement. She has been my pillar of strength, and her presence in my life has been a driving force that kept me going during the challenging times.

VITA

2016	B. S. in Mathematics, University of Liverpool
2016-2023	Graduate Teaching Assistant and Graduate Student Researcher, University of California San Diego
2018	M. S. in Computational Science, University of California San Diego
2021	Software Engineer Intern, AI Algorithm, Kneron Inc.
2022	Applied Scientist PhD Intern, Uber Technologies Inc.
2023	Ph. D. in Mathematics, University of California San Diego

ABSTRACT OF THE DISSERTATION

Cubic Regularization Algorithms for Unconstrained and Constrained Optimization

by

Ziyan Zhu

Doctor of Philosophy in Mathematics

University of California San Diego, 2023

Professor Philip E. Gill, Chair

This dissertation focuses on cubic regularization methods for the globalization of Newton's method, with applications in unconstrained and constrained optimization. In recent years, cubic regularization algorithms have emerged as popular alternatives to trust-region and line-search methods for unconstrained optimization. The goal of this research is to tackle some of the challenges associated with cubic regularization and extend the methods to solve constrained problems. The first part of the dissertation is dedicated to enhancing the efficiency of cubic regularization methods for unconstrained optimization without sacrificing their favorable convergence properties. A nonmonotone adaptive cubic regularization approach is proposed that combines adaptive cubic regularization with a line search. In particular, a sufficient decrease in the objective function is obtained by performing a nonmonotone line-search based on satisfying certain strong Wolfe conditions. This is an alternative to repeatedly solving the cubic subproblem with varying regularization parameters and requires lower computational cost and fewer iterations. In addition, two hybrid algorithms are developed that substitute cubic regularization with Newton's method when the objective function is well-behaved or a sufficient descent direction is readily obtainable from a conventional Newton method. By judiciously determining when to utilize cubic regularization, the efficiency of Newton's method can be balanced against the robustness of cubic regularization. In the second part of the dissertation, a novel primal-dual interior-point method for general nonlinear constrained optimization is proposed that minimizes a sequence of shifted primal-dual penalty-barrier functions. The method combines cubic regularization and a line-search to ensure global convergence. The proposed method calculates the cubic regularized step by factoring a sequence of matrices with diagonally-modified primal-dual structure, enabling the use of off-the-shelf linear equation software. This approach allows the extension of the method to large-scale problems while maintaining computational efficiency. Finally, the performance of the proposed algorithms for both unconstrained and constrained optimization is illustrated by extensive numerical results obtained from problems in the CUTEst test collection.

Chapter 1

Introduction

1.1 Overview

Mathematical optimization is a crucial tool across various domains. It aims to find the best values for variables that maximize or minimize a given function. Optimization problems arise in a wide range of quantitative disciplines, such as computer science, engineering, operations research, and economics. The development of solution techniques has been a topic of interest in mathematics for centuries. In an optimization problem, there are independent variables or parameters, often accompanied by constraints that impose conditions or limitations on the values of the variables. Another crucial component of an optimization problem is the objective function, which serves as a quantitative measurement of the performance of the system under study and is dependent on the values of the variables. The goal is to determine a collection of acceptable values for variables that optimize, i.e., minimize or maximize, the objective function. As maximizing a function is equivalent to minimizing its negative value, we can, without loss of generality, focus on minimizing the objective function.

In its most general form, an optimization problem can be expressed as:

$$\underset{x \in \Omega}{\text{minimize}} \quad f(x),$$

where x represents an n-dimensional vector with components x_1, x_2, $\ldots$, x_n, and Ω denotes the set of allowed values for x. If there are no restrictions on the values of the variables x_1, x_2, $\ldots$, x_n, meaning $\Omega = \mathbb{R}^n$, these problems are considered as unconstrained optimization problems. Unconstrained optimization problems are typically classified based on the properties of the function f. In this thesis, our focus lies on methods for solving general nonlinear unconstrained problems, assuming that f is twice-continuously differentiable. Trust-region and line-search methods are two prevalent second-order approaches for nonlinear unconstrained optimization. The use of a cubic overestimator of the objective function as a regularization technique offers a third alternative. Cubic regularization has gained attention over the past decade due to its appealing theoretical properties and robust numerical performance (see Section 2.3 and Section 5.2). In Chapter 3, we propose and analyze two modifications to the cubic regularization method with the aim of boosting its numerical efficiency and robustness for solving unconstrained problems.

In other cases, the feasible region Ω is defined algebraically through a set of functional equalities or inequalities. These problems are called constrained optimization problems and can be written in the form

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}} && f(x) \\ &\text{subject to} && x_l \le x \le x_u, \quad c_l \le c(x) \le c_u, \end{aligned}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}^m$, $f : \mathbb{R}^n \mapsto \mathbb{R}$, and x_l, x_u, c_l and c_u are constant vectors of lower and upper bounds. As in the unconstrained case, we assume $f(x)$ and $c(x)$ are twice continuously differentiable. The class of primal-dual path-following interior-point methods is considered one of the most successful approaches for solving constrained optimization problems. These methods leverage the properties of both the primal and dual formulations of the problem to achieve efficient and accurate solutions (see Section 2.4). In Chapter 4, a primal-dual cubic regularization algorithm for constrained optimization is proposed that is based on minimizing a shifted primal-dual penalty-barrier function.

1.2 Contributions of This Dissertation

The research presented in this dissertation is motivated by our interest in regularization methods for the globalization of Newton's method. In recent years, cubic regularization algorithms have emerged as alternatives to trust-region and line search approaches. These algorithms employ a strategy that involves determining an approximate global minimizer of a cubic overestimation of the objective function. The goal of this thesis is to tackle the some of the challenges that arise in cubic regularization algorithms and extend these algorithms to primal-dual interior-point methods for solving constrained optimization problems.

In general, the primary computational expense of the cubic regularization framework comes from solving the cubic regularized subproblem, which requires solving one or more linear systems either exactly or through an iterative process. To address this issue, we present a nonmonotone adaptive cubic regularization method that combines adaptive cubic regularization with line-search techniques. In particular, the proposed algorithm performs a nonmonotone line-search based on satisfying certain strong Wolfe conditions along the direction of the rejected trial step to obtain a sufficient decrease step. This strategy circumvents the need for repeatedly solving the cubic subproblem with varying regularization parameters to acquire a trial step that satisfies the sufficient decrease condition. As a result, the method leads to fewer iterations and ultimately decreases the overall computational cost, especially when the function evaluation is not computationally expensive.

Another proposed improvement to the cubic regularization method is a hybrid approach that combines Newton's method with the cubic regularization method. While the Newton method has a fast rate of convergence, it can be impractical when the Hessian is not positive definite or nearly singular. On the other hand, the cubic regularization method can improve the robustness and efficiency of the optimization process by adding a regularization term to the objective function, but it requires solving computationally expensive cubic subproblems. To address these issues, we proposed two variants of a hybrid algorithm that combines the benefits of Newton's method and the cubic regularization method. The first variant of the proposed hybrid algorithm replaces the cubic step with a Newton step when the objective function is well-behaved and the Hessian is positive definite. In this approach, cubic regularization is used when necessary — such as when the objective function is non-convex or poorly conditioned at a given iteration. The second variant of the hybrid algorithm relaxes the conditions for using the Newton direction to replace the cubic regularized step. Specifically, if the directional derivative of the Newton direction and the objective gradient at the current iterate is bounded away from zero, the trial step is derived from the Newton direction and the sign of the directional derivative. A Wolfe line-search is then applied to the trial step to ensure global convergence. Cubic regularization is used only when the Newton direction is nearly orthogonal to the objective, i.e., the corresponding directional derivative is close to zero. The hybrid approach takes advantage of the strengths of both methods and reduces the number of linear systems that need to be solved, resulting in a significant reduction in computational effort.

For general nonlinearly constrained optimization, we present a new primal-dual interior-point method that is based on finding an approximate solution of a sequence of unconstrained subproblems parameterized by some scalar parameters. Each subproblem is solved using a second-derivative Newton-type method that employs a combined cubic regularization and line-search strategy to ensure global convergence. One of the advantages of the proposed method is that the cubic regularized step can be computed by factorizing a sequence of matrices with diagonally-modified primal-dual structure, where the inertia of these matrices can be determined without recourse to a special factorization method. This allows the use of off-the-shelf linear system software, making it possible to extend the method to large-scale problems.

This dissertation is organized as follows. Chapter 2 provides a comprehensive literature review and essential background information that forms the basis for all subsequent chapters. In Chapter 3, we propose and investigate two modifications to the cubic regularization method, nonmonotone adaptive cubic regularization method and hybrid approach of cubic regularization and Newton' method, aimed at boosting numerical efficiency and robustness for solving unconstrained problems. Chapter 4 explores the application of cubic regularization method within the context of primal-dual interior-point methods to address issues such as ill-conditioning and non-convexity in constrained optimization. Lastly, in Chapter 5, we present an extensive analysis of the numerical results obtained through the proposed algorithms evaluated using the CUTEst test collection for unconstrained and constrained optimization.

1.3 Notation

The majority of the notation used in this dissertation is consistent with the conventions of the standard optimization literature. The objective function is denoted by $f(x)$, with its gradient and Hessian represented as $g(x)$ and $H(x)$, respectively. The constraint function is denoted by $c(x)$, and its Jacobian matrix is represented as $J(x)$. The i-th row of the Jacobian matrix is defined by $\nabla c_i(x)^{\mathrm{T}}$.

In this dissertation, subscripts serve to reference both vector indices and iterates, with the specific meaning determined by the context. For instance, x_k represents the k-th iterate in the sequence $\{x_k\}$, while f_k denotes $f(x_k)$. Throughout, e_i denotes the i-th standard basis vector, corresponding to a vector in an n-dimensional Euclidean space characterized by a one in the i-th position and zeros in all other positions.

Unless explicitly stated otherwise, the notation $\|\cdot\|$ represents the vector two-norm or the corresponding induced matrix two-norm. The spectrum of a possibly unsymmetric matrix A is denoted by $\mathrm{eig}(A)$. The inertia of a real symmetric matrix A, denoted by $\mathrm{In}(A)$, is the integer triplet (n_+, n_-, n_0) that indicates the number of positive, negative, and zero eigenvalues of A. The i-th eigenvalue of a symmetric matrix A, with eigenvalues arranged in descending order, is represented by λ_i, i.e., $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Given vectors x_1 and x_2, the column vector composed of the elements of x_1 augmented by the elements of x_2 is denoted by (x_1, x_2). The vector e denotes the vector of ones, with its dimension determined by the context.

Chapter 2

Background

This chapter incorporates background information for the succeeding chapters. Section 2.1 is dedicated to optimality conditions for unconstrained optimization problems and nonlinear mixed constrained optimization problems, where the proofs will be omitted. Section 2.2 provides an introduction to a trust-region method for unconstrained optimization problems, for comparison with the cubic regularization method. Section 2.3 contains background on adaptive cubic regularization methods on which this thesis is based. Section 2.4 describes a shifted primal-dual penalty-barrier function that is a basis of the proposed primal-dual cubic regularization method in Chapter 4.

2.1 Optimality Conditions

2.1.1 Unconstrained Optimization Problems

Unconstrained optimization focuses on the minimization of a scalar-valued function, denoted by f, without constraints on its input values of x. This section reviews the necessary and sufficient conditions for a point to be a local minimizer of f.

First, we provide a version of Definition 2.1.1 for different types of local unconstrained minimizer.

Definition 2.1.1 (Local unconstrained minimizer; strict unconstrained minimizer; weak minimizer)**.** *Given $f : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$, x^* is a* local unconstrained minimizer *if there exits* *an open ball $\mathcal{B}(x^*, \delta)$ such that $\mathcal{B}(x^*, \delta) \subset \mathcal{D}$ and*

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

If the inequality is strict for all $x \in \mathcal{B}(x^*, \delta)$ and $x \neq x^*$, then x^* is said to be a strict minimizer*. An unconstrained minimizer is a* weak unconstrained minimizer *if it is not a strict unconstrained minimizer. If x^* is the only unconstrained minimizer in $\mathcal{B}(x^*, \delta)$, then x^* is a* isolated unconstrained minimizer*.*

A first-order necessary condition is provided in Theorem 2.1.1 that must hold if x^* is an unconstrained minimizer and f is differentiable at x^*.

Theorem 2.1.1 (First-order necessary condition for an unconstrained minimizer)**.** *If $f : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ is differentiable at a local unconstrained minimizer x^*, then $\nabla f(x^*) = 0$.*

Assuming that f has a second derivative near x^*, Theorem 2.1.2 establishes second-order necessary conditions for x^* to be an unconstrained minimizer.

Theorem 2.1.2 (Second-order necessary condition for an unconstrained minimizer)**.** *If $f : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ is twice differentiable at a local unconstrained minimizer x^*, then $\nabla f(x^*) = 0$ and the Hessian matrix $\nabla^2 f(x^*)$ is positive semidefinite.*

The next theorem shows that if x^* is a stationary point where f has a second derivative and the Hessian matrix $\nabla^2 f(x^*)$ is positive definite, then x^* must be an isolated unconstrained minimizer.

Theorem 2.1.3 (Second-order sufficient condition for an unconstrained minimizer)**.** *Given $f : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$, let x^* be an interior point of $\mathcal{D}$ and assume that f has a second derivative at x^*. If $\nabla f(x^*) = 0$ and the Hessian matrix $\nabla^2 f(x^*)$ is positive definite, then x^* is an isolated (and strict) unconstrained minimizer.*

These optimality conditions not only can identify a solution, but also are helpful in designing optimization algorithms. From Definition 2.1.1, if $\bar{x}$ is an interior point of $\mathcal{D}$ that is not yet an unconstrained minimizer, every neighborhood of $\bar{x}$ must contain points whose values of f are strictly less than $f(\bar{x})$. Thus there must exist at least one direction along which one can move away from a non-minimizer and strictly reduce f. To make this precise, the following definition describes a *direction of decrease* as a vector p with the property that all sufficiently small steps along p away from a given point produce strictly lower values of f.

Definition 2.1.2 (Direction of decrease)**.** *Let $f : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ be continuous on a convex set $\mathcal{D}$. A vector $p \in \mathbb{R}^n$ is a* direction of decrease *for f at an interior point $x \in \mathcal{D}$ if there exists a positive $\widehat{\alpha}$ such that $x + \widehat{\alpha} p \in \mathcal{D}$ and $f(x + \alpha p) < f(x)$ for all $\alpha \in (0, \widehat{\alpha})$.*

The following definitions define two directions that are helpful for the detection of directions of decrease.

Definition 2.1.3 (Descent direction)**.** *Let $f : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable at x, an interior point of $\mathcal{D}$. The vector p is a* descent direction *for f at x if $\nabla f(x)^{\mathrm{T}} p < 0$.*

Definition 2.1.4 (Direction of negative curvature)**.** *Let $f : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ have a second derivative at x, an interior point of $\mathcal{D}$. The vector p is a* direction of negative curvature *for f at x if $p^{\mathrm{T}} \nabla^2 f(x) p < 0$.*

The next theorem states verifiable conditions for characterizing directions of decrease in two scenarios: when f is continuously differentiable and when f has a second derivative.

Theorem 2.1.4 (Existence of a direction of decrease)**.** *Given $f : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ on a convex set $\mathcal{D}$, assume that f is continuously differentiable on $\mathcal{D}$, and let x be an interior point of $\mathcal{D}$.*

- *If the vector p satisfies $\nabla f(x)^{\mathrm{T}} p < 0$, then p is a direction of decrease for f at x.*

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

2.1.2 Constrained Optimization Problems

This section reviews the constraint qualification and optimality conditions for a nonlinear mixed-constraint optimization problem written in the form:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; & f(x) \\ \text{subject to} \;\; & c_i(x) = 0, \;\; i \in \mathcal{E} \\ & c_j(x) \ge 0, \;\; j \in \mathcal{I}, \end{aligned} \tag{2.1.1}$$

where $c_i(x)$ and $c_j(x)$ are nonlinear constraint functions, and $\mathcal{E}$ and $\mathcal{I}$ are nonintersecting index sets, representing the equality and inequality components of the nonlinear constraints respectively. Let $c_{\mathcal{E}}(x)$ denote the vector of components $c_i(x)$ with $i \in \mathcal{E}$, and $c_{\mathcal{I}}(x)$ denote the vector of components $c_j(x)$ with $j \in \mathcal{I}$. The following definition provides a definition for a first-order Karush-Kuhn-Tucker (KKT) point of the problem (2.1.1).

Definition 2.1.5 (First-order KKT point)**.** *The first-order KKT conditions for problem* (2.1.1) *hold at the point x^*, or, equivalently, x^* is a first-order KKT point, if there exists a Lagrange multiplier vector y^* such that*

$$c_{\mathcal{E}}(x^*) = 0 \;\; \textit{and} \;\; c_{\mathcal{I}}(x^*) \ge 0, 0 \qquad (\textit{feasibility}), \tag{2.1.2}$$

$$\nabla f(x^*) = J(x^*)^{\mathrm{T}} y^*, \qquad (\textit{stationarity}), \tag{2.1.3}$$

$$y^*_{\mathcal{I}} \ge 0, \qquad (\textit{nonnegativity}), \;\; \textit{and} \tag{2.1.4}$$

$$c_{\mathcal{I}}(x^*) \cdot y^*_{\mathcal{I}} = 0, \qquad (\textit{complementarity}). \tag{2.1.5}$$

The first-order KKT conditions may be written compactly as $F(x, y) = 0$, $c_{\mathcal{I}}(x^*) \ge$

0, $y_{\mathcal{I}} \geq 0$, with

$$F(x,y) = \begin{pmatrix} \nabla f(x) - J^{\mathrm{T}}(x)y \\ c_{\mathcal{I}}(x) \cdot y_{\mathcal{I}} \\ c_{\mathcal{E}} \end{pmatrix}.$$

The KKT conditions rely on the properties of the constraint linearizations, so they are necessary optimality conditions only if certain constraint regularity conditions (i.e., constraint qualifications) are satisfied.

Definition 2.1.6 (Active, inactive, and violated constraints)**.** *For the inequality constraints $c_{\mathcal{I}}(x) \geq 0$, the i-th constraint is said to be active at $\bar{x}$ if $c_i(\bar{x}) = 0$, inactive if $c_i(\bar{x}) > 0$ and violated if $c_i(\bar{x}) < 0$. For the equality constraints $c_{\mathcal{E}}(x) = 0$, the i-th constraint is satisfied at $\bar{x}$ if $c_i(\bar{x}) = 0$ and violated at $\bar{x}$ if $c_i(\bar{x}) \neq 0$. The set of active inequality constraints at $\bar{x}$ is denoted by $\mathcal{A}_A(\bar{x})$, i.e., $\mathcal{A}_A(\bar{x}) = \{\, i \in \mathcal{I} : c_i(\bar{x}) = 0 \,\}$.*

It is important to note that at any feasible point of the problem (2.1.1), all equality constraints are satisfied, that is, $c_i(\bar{x}) = 0$ for $i \in \mathcal{E} \cup \mathcal{A}_A(\bar{x})$. With this in mind, we proceed to formulate the two main constraint qualifications to apply to problems with mixed constraints.

Definition 2.1.7 (LICQ for mixed constraints)**.** *The linear independence constraint qualification holds at the feasible point $\bar{x}$ of problem* (2.1.1) *if the constraint gradients $\nabla c_i(\bar{x})$, $i \in \mathcal{E} \cup \mathcal{A}_A(\bar{x})$ are linearly independent.*

Definition 2.1.8 (MFCQ for mixed constraints)**.** *The Mangasarian–Fromovitz constraint qualification holds at the feasible point $\bar{x}$ of problem* (2.1.1) *if the gradients of the equality constraints at $\bar{x}$, $\nabla c_i(\bar{x})$, $i \in \mathcal{E}$, are linearly independent and if there exists a vector p such that $\nabla c_i(\bar{x})^{\mathrm{T}} p > 0$ for all $i \in \mathcal{A}_A(\bar{x})$ and $\nabla c_i(\bar{x})^{\mathrm{T}} p = 0$ for all $i \in \mathcal{E}$.*

The following result presents the necessary conditions for optimality under the assumption that a constraint qualification holds.

Theorem 2.1.5 (First-order necessary conditions for mixed constraints)**.** *If x^* is a local minimizer of problem* (2.1.1) *and the MFCQ holds at x^*, then x^* must be a KKT point.*

Depending on the nature of x^*, there may be an infinite number of multipliers satisfying the stationarity condition (2.1.3). The set of multipliers that satisfy the KKT conditions is defined as follows.

Definition 2.1.9. *(Acceptable Lagrange multipliers for mixed constraints). Given a KKT point x^* for problem* (2.1.1)*, the set of acceptable multipliers is defined as*

$$\mathcal{Y}(x^*) \overset{\text{def}}{=} \left\{ y \in \mathbb{R}^m : \nabla f(x^*) = J(x^*)^{\mathrm{T}} y, y_{\mathcal{I}} \geq 0, \ \ \textit{and} \ \ y_{\mathcal{I}} \cdot c_{\mathcal{I}}(x^*) = 0 \right\}.$$

With the definition of $\mathcal{Y}(x^*)$, second-order necessary conditions for optimality can be stated when the LICQ holds.

Theorem 2.1.6 (Second-order necessary conditions)**.** *Suppose that x^* is a local minimizer of problem* (2.1.1) *at which the LICQ holds. Then there is a vector $y^* \in \mathcal{Y}(x^*)$ and $p^{\mathrm{T}} H(x^*, y^*) p \geq 0$ for all p satisfying $\nabla f(x^*)^{\mathrm{T}} p = 0$, $J_{\mathcal{E}}(x^*) p = 0$ and $J_{\mathcal{A}}(x^*) p \geq 0$.*

The following result demonstrates the second-order sufficient optimality conditions for problem (2.1.1).

Theorem 2.1.7 (Second-order sufficient conditions)**.** *Let x^* denote a KKT point of problem* (2.1.1)*. We say that second-order sufficient conditions hold at x^* if for every Lagrange multiplier y satisfying $y_{\mathcal{I}} \geq 0$, $c_{\mathcal{I}}^* \cdot y_{\mathcal{I}} = 0$, and $\nabla f(x^*) = J(x^*)^{\mathrm{T}} y^*$, there exists $\omega > 0$ such that $p^{\mathrm{T}} H(x^*, y^*) p \geq \omega \|p\|^2$ for all nonzero p such that $\nabla f(x^*)^{\mathrm{T}} p = 0$, $J_{\mathcal{E}}(x^*) p = 0$ and $J_{\mathcal{A}}(x^*) p \geq 0$.*

Theorem 2.1.7 is a useful tool for characterizing a local constrained minimizer without the need for a constraint qualification. However, to establish that x^* is an isolated local constrained minimizer, a stronger result, the MFCQ constraint qualification is required, as shown in the following theorem.

Theorem 2.1.8 (Sufficient conditions for an isolated solution). *The point x^* is an isolated local constrained minimizer of problem* (2.1.1) *if*

- *x^* is a KKT point, i.e., $c_{\mathcal{I}}^* \geq 0$, $c_{\mathcal{E}}^* = 0$, and there exists a nonempty set $\mathcal{Y}$ of multipliers y satisfying $y_{\mathcal{I}} \geq 0$, $c_{\mathcal{I}}^* \cdot y_{\mathcal{I}} = 0$, and $\nabla f(x^*) = J(x^*)^{\mathrm{T}} y$;*
- *the MFCQ holds at x^*;*
- *for all $y \in \mathcal{Y}$ and all nonzero p satisfying $\nabla f(x^*)^{\mathrm{T}} p = 0$, $J_{\mathcal{E}}(x^*)p = 0$ and $J_{\mathcal{A}}(x^*)p \geq 0$, there exists $\omega > 0$ such that $p^{\mathrm{T}} H(x^*, y^*)p \geq \omega \|p\|^2$.*

The following result provides an alternative criterion for identifying an isolated local constrained minimizer under the LICQ constraint qualification.

Theorem 2.1.9 (Sufficient conditions for an isolated solution). *The point x^* is an isolated local constrained minimizer of problem* (2.1.1) *if*

- *x^* is a KKT point and strict complementarity holds, i.e., the (necessarily unique) multiplier y^* has the property that $[y_{\mathcal{A}}^*]_i > 0$ for all $i \in \mathcal{A}_A(x^*)$;*
- *x^* is a feasible and LICQ holds at x^*.*
- *for all nonzero p such that $J_{\mathcal{A}}(x^*)p \geq 0$, there exists $\omega > 0$ such that $p^{\mathrm{T}} H(x^*, y^*)p \geq \omega \|p\|^2$.*

2.2 Trust-Region Methods

Trust-region methods are a popular class of algorithms for unconstrained optimization and are effective for globalizing Newton-like iterations. These methods use a quadratic model of the objective in a region around the current iterate, and solve a constrained subproblem to determine the next iterate. The trust-region subproblem involves minimizing the local quadratic function subject to a trust-region constraint that restricts the length of the step. By controlling the step size, trust-region methods aim to balance the need for making progress towards optimality with the need to stay within a region where the local quadratic model is accurate. In this section, we provide a brief overview of trust-region methods, which will serve as the baseline for evaluating and comparing the ARC methods discussed in Section 5.2.

Algorithm 2.1 outlines a basic trust-region algorithm that can be used to solve unconstrained optimization problems with an objective function f. Trust-region methods

Algorithm 2.1. Schematic outline of basic trust-region algorithm.

```
1: function TRUST_REGION_ALGORITHM
2:     Initialization: Given $x_0$, $\gamma_2 > 1 > \gamma_1 > 0$, $1 > \eta_2 > \eta_1 > 0$, $k = 0$, $\delta_k = 1$.
3:     while not converged do
4:         Compute a step $p_k$ as an approximate solution of the subproblem (2.2.1).
5:         Compute the ratio $\rho_k = \dfrac{f(x_k + p_k) - f(x_k)}{Q(p_k) - f(x_k)}$.
6:         if $\rho_k \ge \eta_1$ then
7:             $x_{k+1} = x_k + p_k$.
8:             if $\rho_k \ge \eta_2$ then
9:                 $\delta_{k+1} = \max\{\, \delta_k, \gamma_2 \|p_k\| \,\}$.
10:            else
11:                $\delta_{k+1} = \delta_k$.
12:            end if
13:        else
14:            $x_{k+1} = x_k$.
15:            $\delta_{k+1} = \gamma_1 \|p_k\|$.
16:        end if
17:        $k = k + 1$.
18:    end while
19: end function
```

explicitly impose a constraint on the length of the step by defining p_k as an approximate solution of the constrained minimization problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; \mathcal{Q}_k(s) = f(x_k) + \nabla f(x_k)^{\mathrm{T}} s + \tfrac{1}{2} s^{\mathrm{T}} B_k s \quad \text{subject to} \quad \|s\| \le \delta_k, \tag{2.2.1}$$

where $\delta_k > 0$ is the trust-region radius and B_k is either the exact or an approximate

Hessian evaluated at $x = x_k$. After solving the subproblem (2.2.1) for the trust-region step p_k, the ratio of actual reduction to the predicted reduction in the objective function f is computed as

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

If the actual reduction is not less than a certain fraction η_1 of the predicted reduction, i.e., $\rho_k \geq \eta_1$, then the new point x_{k+1} is set to $x_k + p_k$. However, if the test fails, i.e., $\rho_k < \eta_1$, then $x_{k+1} = x_k$, the trust-region radius is decreased by a contraction factor γ_2 and the subproblem is solved again. This approach is motivated by the idea that the value of $Q(p_k)$ for the next subproblem will provide a better estimate of $f(x_k + p_k)$. In practice, it may take several adjustments of the trust-region radius before the predicted and actual reductions become comparable.

The main effort associated with the trust-region method is the calculation of an approximate solution p_k of the trust-region subproblem (2.2.1). The following lemma given by Moré and Sorensen [34] provides the theoretical basis for solving trust-region subproblem and measuring the quality of an approximate solution.

Lemma 2.2.1 (Lemma 4.1.1 [34]). *Let δ be a given positive constant. A vector s^* is a global minimizer of the trust-region subproblem $\mathcal{Q}_k(s)$ if and only if $\|s^*\| \leq \delta$ and there exists a unique $\lambda^* \geq 0$ such that*

$$(B_k + \lambda^* I)s^* = -\nabla f(x_k), \quad \lambda^*(\delta - \|s^*\|) = 0, \tag{2.2.2}$$

with $B_k + \lambda^* I$ positive semidefinite. Moreover, if $B + \lambda^* I$ is positive definite, then the global minimizer s^* is unique.

Mor'e and Sorensen proposed an iterative method for solving the trust-region subproblem (2.2.1) based on Lemma 2.2.1. The algorithm aims to find an approximate solution p_k satisfying

$$\mathcal{Q}(s) \leq \tau \mathcal{Q}^*,$$

where $\mathcal{Q}^*$ is the optimal value of the trust-region subproblem and τ is a scalar such that $0 < \tau < 1$. To achieve this, a safeguarded Newton iteration is applied to find a root of (2.2.2) and ensure that λ remains within the interval $[0, \infty)$. This approach is guaranteed to find a nearly optimal solution in a finite number of iterations.

In a paper by Gertz [33], a line search based on satisfying the Wolfe conditions is combined with a conventional trust-region method for unconstrained minimization. This modification preserves the fast convergence and stability of trust-region methods while reducing the average cost per iteration. Additionally, Gertz proposed a "biased" updating strategy for the trust region radius δ_k based on the line-search step length, thereby avoiding unnecessarily small choices of δ_{k+1}. Numerical experiments demonstrate that the biased Wolfe trust-region method is both efficient and robust in practice, and, as far as we know, exhibits the best numerical performance of any trust-region method. Consequently, this algorithm will be used as a benchmark for comparing variants of the ARC algorithm (see Section 5.2).

2.3 Cubic Regularization Methods

This section provides a comprehensive summary of the inspiration and latest developments in the cubic regularization method. It serves as the fundamental context for this thesis. Specifically, Section 2.3.1 examines the inspiration behind cubic regularization and provides a survey of the relevant literature. Section 2.3.2 discusses an adaptive cubic regularization approach and its theoretical properties. Finally, Section 2.3.3 outlines the algorithm for solving the cubic subproblem.

2.3.1 Introduction

Consider the unconstrained nonlinear programming problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth function. The well known globally convergent algorithms, such as line-search and trust-region methods, are commonly used to solve unconstrained optimization. Recently, cubic regularization methods have become alternatives to these methods for globalizing Newton-like methods. The main idea of cubic regularization algorithms is using a cubic over-estimator of the objective function at current iterate as a regularization technique to compute the search direction for the next step. The method has been shown to exhibit both excellent local and global convergence properties, which makes it well-suited for solving a wide range of optimization problems. Specifically, suppose that $f(x)$ has gradient $\nabla f(x)$ and globally Lipschitz continuous Hessian $\nabla^2 f(x)$ with ℓ_2 norm Lipschitz constant $2L$:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \le 2L\|x-y\|, \quad \forall x, y \in \mathbb{R}^n.$$

Then, by Taylor's theorem, we get the following inequality:

$$\begin{aligned} f(x+p) &= f(x) + \nabla f(x)^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} \nabla^2 f(x) p + \int_0^1 (1-t) p^{\mathrm{T}} (\nabla^2 f(x+tp) - \nabla^2 f(x)) p \, dt \\ &\le f(x) + \nabla f(x)^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} \nabla^2 f(x) p + \frac{1}{3} L \|p\|^3 \overset{\text{def}}{=} m_k^C(p). \end{aligned} \tag{2.3.1}$$

As long as p_k minimizes the overestimation model $m_k^C(p)$, i.e.,

$$f(x_k + p_k) \le m_k^C(p_k) < m_k^C(0) = f(x_k),$$

the next iterate $x_{k+1} = x_k + p_k$ reduces the objective value.

Griewank [29] first considered computing the step by minimizing $m_k^C(p)$ in a technical report. The Lipschitz constant $2L$ is replaced by a regularization parameter σ_k that regularizes the Newton local quadratic model. In particular, Griewank proposed the use of the local model

$$m_k^G(p) \overset{\text{def}}{=} f(x_k) + \nabla f(x_k)^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} \nabla^2 f(x_k) p + \frac{1}{3}\sigma_k \|p\|_G^3,$$

where $\sigma_k\| \cdot \|_G^3$ is chosen dynamically to ensure the overestimation property. Griewank also established global convergence to a second-order stationary point for a method in which a descent direction is generated by computing a second-order minimizer of $m_k^G(p)$.

Recently, Nesterov and Polyak [36] analyzed the iteration complexity for the cubic regularization method in which the step is computed by a global minimizer of the cubic model $m_k^C(p)$ and the Hessian is globally Lipschitz continuous. This assumption allows the derivation of a global iteration-complexity bound of $\mathcal{O}(\epsilon^{-3/2})$ for solving the problem to within a certain level of optimality measured by the parameter $\epsilon \in (0,1)$. This result extends previous work because the method achieves a better global iteration complexity than the steepest-descent method. Moreover, the method can solve nonlinear optimization problems to arbitrary precision under weaker assumptions than other second-order optimization methods such as Newton's method. Global convergence and an asymptotically quadratic rate of convergence were also established.

More recently, Cartis, Gould and Toint [8] proposed a numerically efficient adaptive regularized framework using cubics (ARC) which preserves the good iteration complexity bound analyzed by Nesterov and Polyak [36]. The global and asymptotic convergence results were also proved under weaker assumptions. The specific assumptions that guarantee the local and global convergence properties of the ARC method can vary depending on the specific method. However, the most common assumption is that the objective function is Lipschitz continuous and has a bounded Hessian. These are quite weak assumptions in the context of optimization. Cartis, Gould and Toint consider the model

$$m_k(p) \overset{\text{def}}{=} f(x_k) + \nabla f(x_k)^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} B_k p + \frac{1}{3}\sigma_k \|p\|^3,$$

as an approximation to f at $x = x_k + p$, where B_k is an approximation of the exact Hessian at x_k. Note that, in contrast to line-search methods, the approximated Hessian B_k need not be positive definite. Moreover, the requirement that a global minimizer of the cubic model must be found is relaxed to require only that the step p_k be at least as good as a suitable Cauchy step. This requirement makes the practical implementation of the ARC method feasible. In terms of the complexity bound, Cartis, Gould and Toint [9] show that the ACR method achieves the optimal complexity bound among second-order methods for a class of nonconvex optimization problems known as "well-conditioned" problems. Problems in this class have a curvature condition and a Lipschitz continuity condition on the gradient.

2.3.2 Adaptive Cubic Regularization Methods

For brevity, we denote $f_k = f(x_k)$, $g_k = \nabla f(x_k)$ and $H_k = \nabla^2 f(x_k)$ throughout this section. At each iteration, we define a local quadratic model of f at $x_k + p$:

$$m_k^Q(p) \overset{\text{def}}{=} f_k + g_k^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} B_k p, \tag{2.3.2}$$

where B_k is an approximates the objective Hessian at x_k. The corresponding cubic regularized model with an adaptive regularization parameter $\sigma_k > 0$ is given by

$$m_k(p) \overset{\text{def}}{=} m_k^Q(p) + \frac{1}{3}\sigma_k \|p\|^3 = f_k + g_k^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} B_k p + \frac{1}{3}\sigma_k \|p\|^3, \tag{2.3.3}$$

where $\|\cdot\|$ denotes the ℓ_2 norm unless otherwise specified. The generic adaptive cubic regularization scheme proposed by Cartis, Gould and Toint [8] is summarized in

Algorithm 2.2.

Algorithm 2.2. Schematic outline of a regularization method using cubics.

```
1: function ADAPTIVE_CUBIC_REGULARIZATION
2:     Initialization: Given $x_0$, $\gamma_2 \geq \gamma_1 > 1$, $1 > \eta_2 \geq \eta_1 > 0$, $\sigma_0 > 0$, $k = 0$;
3:     while not converged do
4:         Trial step computation. Compute a step $p_k$ for which
               $$m_k(p_k) \leq m_k(p_k^C); \quad (2.3.4)$$
           where $p_k^C$ is the Cauchy step
               $$p_k^C = -\alpha_k^C g_k \text{ and } \alpha_k^C = \operatorname*{argmin}_{\alpha \in \mathbb{R}_+} m_k(-\alpha_k g_k); \quad (2.3.5)$$
5:         Ratio computation.
               $$\rho_k = \frac{f_k - f(x_k + p_k)}{f_k - m_k(p_k)}; \quad (2.3.6)$$
6:         Step acceptance.
               $$\text{Update} \quad x_{k+1} = \begin{cases} x_k + p_k & \text{if } \rho \geq \eta_1; \\ x_k & \text{otherwise}; \end{cases} \quad (2.3.7)$$
7:         Regularization parameter update.
               $$\text{Update} \quad \sigma_{k+1} \in \begin{cases} (0, \sigma_k] & \text{if } \rho_k > \eta_2, & \text{[very successful iteration]} \\ [\sigma_k, \gamma_1 \sigma_k] & \text{if } \eta_1 \leq \rho_k \leq \eta_2, & \text{[successful iteration]} \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{otherwise.} & \text{[unsuccessful iteration]} \end{cases} \quad (2.3.8)$$
8:         $k = k + 1$;
9:     end while
10: end function
```

At the current iterate x_k, a step satisfying the condition (2.3.4) is computed as an approximate global minimizer of the cubic model $m_k(p)$ in (2.3.3). Calculating the Cauchy step (2.3.5) is computationally inexpensive because it involves minimizing a cubic polynomial along one dimension. It is worth noting that using a more accurate minimizer for the cubic model $m_k(p)$ may result in improved numerical performance. A particular method for solving the cubic regularized model will be discussed in the following section.

The update for the regularization parameter σ_k is determined by the behavior of f near the point x_k along the step p_k. Once the cubic regularized step p_k has been found, the ratio of the actual reduction to the predicted reduction in objective, using the formula for ρ_k defined in (2.3.6), is computed. The step p_k is accepted if the actual reduction in objective, $f(x_k) - f(x_k + p_k)$, is "large enough" in magnitude compared to the predicted reduction, $f(x_k) - m_k(p_k)$. In particular, given a fixed factor η_1, satisfying $0 < \eta_1 < 1$, the ratio ρ_k is required to be at least no smaller than η_1, i.e., $\rho_k \geq \eta_1$. When the step is accepted, the regularization parameter may be remain unchanged or decreased if good enough agreement between model and objective are observed. By contrast, if the test fails, i.e., $\rho_k < \eta_1$, then the step is rejected and $x_{k+1} = x_k$. The regularization parameter is expected to be increased and the cubic subproblem is solved again. This strategy is based on the observation that the regularization parameter σ_k plays a role in estimating the local Lipschitz constant of objective Hessian, which can lead to a local overestimation of f. Increasing the weight σ_k will implicitly decrease the step size as proved in Lemma 2.3.1 in the next section. It may be necessary to solve the subproblem multiple times before the predicted and actual reductions are comparable.

As norms on $\mathbb{R}^n$ are equivalent, it is possible to use a more general norm $\|x\|_M = (x^{\mathrm{T}} M x)^{1/2}$, in place of the ℓ_2 norm in the model $m_k(p)$. Moreover, the positive-definite matrix M can vary with the iteration number k, as long as it is uniformly bounded and positive definite for all k. The convergence properties of the ARC algorithm continue to be valid in this more general setting.

Theoretical Results

The adaptive cubic regularization method exhibits excellent local and global convergence properties under certain weak assumptions, and it achieves the optimal complexity bound among second-order methods shown by Cartis, Gould and Toint [7]. The complexity bound refers to the number of iterations required for the algorithm to converge to an approximate solution that is within a certain level of optimality. This section provides a summary of some promising theoretical results for the adaptive cubic regularization method. The following assumptions will be frequently used in this section.

Assumption 2.3.1. $\|B_k\| \leq \kappa_B$, *for all* $k \leq 0$, *and some* $\kappa_B \geq 0$.

Assumption 2.3.2 (Dennis-Moré condition)**.** *The error in the matrix* B_k *as an approximation of the exact Hessian* H_k *satisfies:*

$$\frac{\|(B_k - H_k)p_k\|}{\|p_k\|} \to 0, \qquad \textit{whenever } \|g_k\| \to 0, \quad k \to \infty.$$

There are several methods for updating the matrix B_k using quasi-Newton techniques that will satisfy the Dennis-Moré condition, as long as certain additional conditions are satisfied [38].

Assumption 2.3.3. *The approximate Hessian* B_k *approaches the exact Hessian* H_k *whenever the iterates approach a first-order critical point, namely*

$$\|H_k - B_k\| \to 0, \qquad \textit{whenever } \|g_k\| \to 0, \quad k \to \infty. \tag{2.3.9}$$

Assumption 2.3.9 can be theoretically guaranteed when the matrix B_k is set to the approximation of H_k obtained through finite differences [38]. This assumption also holds when using a symmetric rank-one approximation method to update B_k, as long as the steps taken are linearly independent [6, 13].

Assumption 2.3.4. *The error of the approximate Hessian* B_k *satisfies:*

$$\|(B_k - H_k)p_k\| \leq C\|p_k\|^2, \qquad \textit{for all } k \geq 0, \textit{and some constant } C > 0.$$

Assumption 2.3.5. *The gradient* g *is uniformly continuous on the sequence of iterates*

$\{ x_k \}$, *namely,*

$$\|g_{l_i} - g_{m_i}\| \to 0, \quad \textit{whenever } \|x_{l_i} - x_{m_i}\| \to 0, \quad i \to \infty$$

where $\{ x_{l_i} \}$ *and* $\{ x_{l_i} \}$ *are the subsequence of* $\{ x_k \}$.

Assumption 2.3.5 holds true if g is uniformly continuous on $\mathbb{R}^n$, or if g is globally Lipschitz continuous on $\{ x_k \}$.

Assumption 2.3.6. *The gradient g is Lipschitz continuous on an open convex set X containing the sequence of iterates $\{ x_k \}$, namely,*

$$\|g_x - g_y\| \leq \kappa_H \|x - y\|, \quad \textit{for all } x, y \in X, \textit{ and some } \kappa_H \geq 0.$$

Assumption 2.3.6 is satisfied if the Hessian $H(x)$ is bounded above on X.

Assumption 2.3.7. *The Hessian H is locally Lipschitz continuous in a neighborhood of a given point x_*, namely,*

$$\|H(x) - H(y)\| \leq L_* \|x - y\|, \textit{ for all } x, y \textit{ sufficiently close to } x^*, \textit{ and } L_* \geq 0.$$

Assumption 2.3.8. *The Hessian H is globally Lipschitz continuous, namely,*

$$\|H(x) - H(y)\| \leq L_* \|x - y\|, \textit{ for all } x, y \in \mathbb{R}^n, \textit{ and } L_* \geq 0.$$

The next lemma shows that, in contrast to the explicitly controlled step size by the trust-region constraint in the trust-region algorithm, the step size is controlled implicitly and nonlinearly in the ARC algorithm.

Lemma 2.3.1 (Lemma 2.2 [8]). *Suppose that* Assumption 2.3.1 *holds and that the step p_k*

satisfies (2.3.4). *Then*

$$||p_k|| \leq 3 \max\left(\frac{\kappa_B}{\sigma_k}, \sqrt{\frac{||g_k||}{\sigma_k}}\right), \quad k \geq 0.$$

The following theorem demonstrates the first-order convergence of the ARC algorithm under relatively weak conditions.

Theorem 2.3.1 (Theorem 2.5 & Corollary 2.6 [8]). *Let* Assumption 2.3.1 *hold. If* $\{ f(x_k) \}_{k=1}^{\infty}$ *is bounded below, then*

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

In addition, suppose that Assumption 2.3.5 *holds, then*

$$\lim_{k\to\infty} \|g_k\| = 0.$$

The next theorem presents conditions that guarantee the limit points of the sequence of iterates are second-order critical points, without requiring the model or the function f to be globally or locally convex at the iterates or their limit points.

Theorem 2.3.2 (Theorem 5.4 [8]). *Suppose that* Assumptions 2.3.1, 2.3.3, 2.3.4, *and* 2.3.5 *and* 2.3.8 *hold. Assume that* $\{ f(x_k) \}$ *is bounded below, and that* $\sigma \geq \sigma_{\min} > 0$ *for* $k \geq 0$. *Also, let* p_k *be the global minimizer of the cubic model* $m_k(p)$. *Then any subsequence of negative leftmost eigenvalues* $\{ \lambda_{\min}(H_k) \}$ *converges to zero as* $k \to \infty$, *and thus*

$$\liminf_{k\to\infty, k\in\mathcal{S}} \lambda_{\min}(H_k) = 0.$$

Furthermore, any limit point of the sequence of iterates $\{ x_k \}$, *if exists, is second-order critical.*

The next result shows that if the solution of the cubic model at each iteration is sufficiently accurate then, under certain assumptions, the ARC algorithm is at least Q-superlinearly convergent.

Corollary 2.3.1 (Corollary 4.8 and Corollary 4.10 [8]). *Assume that* Assumptions 2.3.1, 2.3.2 *and* 2.3.5 *hold, and that f has second-order derivatives. If the solution p_k of cubic model at each iteration satisfies*

$$\|\nabla m_k(p_k)\| \leq \min(\kappa, \|\nabla m_k(0)\|^{\frac{1}{2}})\|g_k\|$$

for some $0 < \kappa \ll 1$, then $g_k \to 0$, and $x_k \to x^*$ at a Q-superlinear rate as $k \to \infty$, i.e.,

$$\lim_{k\to\infty} \frac{\|g_{k+1}\|}{\|g_k\|} = 0, \quad \textit{and} \quad \lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Furthermore, if Assumptions 2.3.6, 2.3.7 *and* 2.3.4 *hold, then $g_k \to 0$, and $x_k \to x^*$ at a Q-quadratic rate as $k \to \infty$.*

The ARC algorithm has a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for first-order optimality within a margin of error of ϵ, as stated in the next corollary.

Corollary 2.3.2 (Corollary 5.3 [9]). *Suppose that* Assumptions 2.3.4, 2.3.6 *and* 2.3.8 *hold, and that $\{ f(x_k) \}$ is bounded below. Let σ is bounded below by some constant $\sigma_{min} > 0$. Given $\epsilon \in (0, 1)$, if the solution p_k of cubic model at each iteration satisfies*

$$\|\nabla m_k(p_k)\| \leq \min(\kappa, \|\nabla m_k(0)\|^{\frac{1}{2}})\|g_k\|$$

for some $0 < \kappa \ll 1$, then there exists a constant c depending on x_0 such that at most

$$\left\lceil \frac{c}{\epsilon^{3/2}} \right\rceil$$

iterations of cubic regularization method are needed to obtain an iterate x_k such that $\|g_k\| \leq \epsilon$.

It is important to keep in mind that this is a worst-case complexity bound, and the actual number of iterations required for the algorithm to converge may be much smaller in practice. The actual time complexity of the algorithm depends on many factors, such as the specific problem being solved, the properties of the data, and the implementation details.

Updating the Regularization Parameter

The regularization parameter plays a crucial role in the performance of the ACR algorithm, and an appropriate choice can lead to fast convergence and good solution quality. The weight σ_k estimates the local Lipschitz constant of the objective's Hessian, and quantifies the discrepancy between the objective function and its second-order Taylor-series approximation. In contrast to the trust-region algorithm, where the step size is controlled explicitly and linearly by the trust-region constraint, the step size in the ARC algorithm is controlled implicitly and non-linearly, as shown in Lemma 2.3.1. An efficient parameter updating strategy was proposed by Gould, Porcelli and Toint [27] based on interpolation techniques.

The ARC algorithm adjusts the step size based on the agreement between the objective function and the local model at each iteration. When the agreement is poor in an unsuccessful iteration, the step size is reduced by increasing σ_k, in order to increase the chances of success in the next iteration. On the other hand, when there is a good agreement between the objective function and the model in a very successful iteration, it indicates that the model is overestimating the objective function value locally. In this case, σ_k is decreased to reduce the gap between the two. This leads to the design of an update strategy that makes use of the overestimation gap between the current objective function $f(x_k + p_k)$ and the current local model value $m_k(p_k)$ to adjust the step size in order to improve the performance of the algorithm.

To avoid the need for additional function evaluations, a cubic function $p_f(\alpha)$, $\alpha \geq 0$, is used to interpolate $f(x_k + \alpha p_k)$. This function is defined such that it satisfies the following conditions: $p_f(0) = f_k$, $p_f'(0) = g_k^{\mathrm{T}} p_k$, $p_f''(0) = p_k^{\mathrm{T}} B_k p_k$ and $p_f(1) = f(x_k + p_k)$. This interpolation function is chosen to satisfy the following conditions:

- $p_f(0) = f_k$, the objective function value at the current point;
- $p_f'(0) = g_k^{\mathrm{T}} p_k$, the the directional derivative of the objective function along the search direction;
- $p_f''(0) = p_k^{\mathrm{T}} B_k p_k$, the curvature of the objective function at the current point;
- $p_f(1) = f(x_k + p_k)$, the objective function value at the next point.

A possible form of $p_f(\alpha)$ is

$$p_f(\alpha) = f_k + g_k^{\mathrm{T}} p_k \alpha + \frac{1}{2} p_k^{\mathrm{T}} B_k p_k \alpha^2 + p_{f_3} \alpha^3,$$

where $p_{f_3} = f(x_k + p_k) - m_k^Q(p_k)$. The quadratic model (2.3.2) along the search direction p_k may be written as

$$q(\alpha) = f_k + g_k^{\mathrm{T}} p_k \alpha + \frac{1}{2} p_k^{\mathrm{T}} B_k p_k \alpha^2,$$

while its regularized cubic counterpart is

$$c(\alpha, \sigma) = q(\alpha) + \frac{\sigma ||p_k||^3}{3} \alpha^3.$$

The current overestimation gap χ_k^f can be calculated by

$$\chi_k^f = c_k(p_k) - f(x_k + p_k). \tag{2.3.10}$$

Note that the model m_k at p_k overestimates $f(x_k + p_k)$, i.e. $\chi_k^f \geq 0$, if and only if $\rho_k \geq 1$.

In the case of a very successful iteration, where the local model overestimates the objective function, the regularization parameter should be decreased. This will reduce the gap χ_k^f by a factor β, where $\beta \in (0,1)$. Here, two cases are considered, $f(x_k + p_k) \geq m_k^Q(p_k)$ and $f(x_k + p_k) < m_k^Q(p_k)$.

In the case where $f(x_k+p_k) \geq m_k^Q(p_k)$, the objective lies between the local quadratic model and the cubic model. We search for α and σ such that

$$\frac{d}{d\alpha} c(\alpha,\sigma) = 0, \tag{2.3.11}$$

$$c(\alpha,\sigma) - p_f(\alpha) = \beta \chi_k^f. \tag{2.3.12}$$

Equation (2.3.11) is used to find the stationary point of the cubic model along the search direction p_k. At the same time, the gap χ_k^f between the objective function and the cubic model is reduced by a factor β as imposed by equation (2.3.12). After some simplifications, the required α satisfies the following cubic equation

$$3\beta\chi_k^f + g_k^{\mathrm{T}} p_k \alpha + p_k^{\mathrm{T}} B_k p_k \alpha^2 + 3p_{f_3}\alpha^3 = 0.$$

The algorithm seeks the root α, which exceeds $\sqrt[3]{\beta}$ by the least, if it exists. The value of σ is updated by using equation (2.3.12), which can be rewritten as

$$\sigma = \sigma_k + 3\frac{\chi_k^f}{\|p_k\|^3}\left(\frac{\beta - \alpha^3}{\alpha^3}\right).$$

If such a root does not exist or if the value of α is too large, the σ will be reduced by a factor $\delta_1 \in (0,1)$.

When $f(x_k + p_k) < m_k^Q(p_k)$, the objective lies below the local quadratic model. In this case, reducing the gap between the cubic model and the objective function may result in reducing it too much. Instead, the algorithm aims to reduce the gap between the cubic model and the quadratic model defined as follows

$$\chi_k^q = c_k(p_k) - q_k(p_k), \tag{2.3.13}$$

and searches for α and σ satisfying

$$\frac{d}{d\alpha} c(\alpha, \sigma) = 0, \tag{2.3.14}$$

$$c(\alpha, \sigma) - q(\alpha) = \beta \chi_q^f. \tag{2.3.15}$$

Equation (2.3.14) finds a stationary point of the cubic model with respect to α along the search direction, and equation (2.3.15) reduces the gap between the cubic model and the quadratic model by a factor β. After some simplifications, the required α satisfies the following quadratic equation

$$3\beta\chi_k^q + g_k^{\mathrm{T}} p_k \alpha + p_k^{\mathrm{T}} B_k p_k \alpha^2 = 0.$$

Provided it exists, the algorithm pursues the root α which exceeds $\sqrt[3]{\beta}$ by the least, and recovers σ by (2.3.15), which can be rewritten as

$$\sigma = \frac{\beta}{\alpha^3} \sigma_k.$$

If such a root does not exist or if the value of α is too large, the algorithm will simply decrease σ by a factor $\delta_1 \in (0, 1)$.

In the case of a very unsuccessful iteration ($\rho_k < 0$), where the objective function value at the next point is not much better than the current point, the regularization parameter should be increased. The algorithm searches for αp_k and σ such that the next iteration is expected to be at least a successful iteration. This is done by finding α and σ

that satisfy the following conditions:

$$\frac{d}{d\alpha}c(\alpha,\sigma) = 0, \tag{2.3.16}$$

$$f - p_f(\alpha) = \eta(f - c(\alpha,\sigma)), \tag{2.3.17}$$

for some $\eta \in [\eta_1, 1)$. Equation (2.3.16) finds α which is a stationary point of the cubic model along the search direction, and equation (2.3.17) increases the gap between the objective function and the cubic model by a factor η where $\eta \in [\eta_1, 1)$. After some simplifications, the required α satisfies the following quadratic equation

$$2(3-2\eta)g_k^{\mathrm{T}}p_k + (3-\eta)p_k^{\mathrm{T}}B_kp_k\alpha + 6p_{f_3}\alpha^2 = 0.$$

If it exists, the algorithm explores the positive root α, and recover σ by (2.3.17), which can be rewritten as

$$\sigma = \frac{-g_k^{\mathrm{T}}p_k - p_k^{\mathrm{T}}B_kp_k\alpha}{\alpha^2\|p_k\|^3}.$$

Otherwise, if the positive root α does not exists, the algorithm increases σ by a factor $\delta_3 > 1$.

For other situations, the algorithm follows the usual updating rules as specified in equation (2.3.8).

2.3.3 Solving the Cubic Regularized Subproblem

The practicality of adaptive cubic regularization algorithm depends on the efficiency of solving the cubic regularization subproblem at each iteration. Because of the similarity between the optimality conditions of trust-region subproblem and cubic regularization subproblem, the strategy proposed by Moré and Sorensen [34] could be applied to find a suitable approximate global minimizer of the cubic model. More recently, Gould, Robinson and Thorne [28] proposed an enhancement to Moré-Sorensen algorithm using higher order polynomial approximations to achieve fast convergence rate. Throughout this section, the (outer) iteration subscript k is dropped for simplicity.

Consider the cubic regularization subproblem defined by

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ \mathcal{C}(s) \overset{\text{def}}{=} g^{\mathrm{T}}s + \tfrac{1}{2}s^{\mathrm{T}}Bs + \frac{1}{3}\sigma\|s\|_M^3, \tag{2.3.18}$$

where the M-norm of x is $\|x\|_M = \sqrt{x^{\mathrm{T}}Mx}$. The following theorem provides the necessary and sufficient optimality conditions for the subproblem (2.3.18) and measures the quality of an approximate minimizer.

Theorem 2.3.3 (Theorem 2 [28]). *Any s^* is a global minimizer of the cubic model $\mathcal{C}(s)$ over $\mathbb{R}^n$ if and only if there exists a unique $\lambda^* \geq 0$ such that*

$$(B + \lambda^* M)s^* = -g, \qquad \frac{\lambda^*}{\sigma} = \|s^*\|_M,$$

with $B + \lambda^* M$ positive semidefinite. For any global minimizer s^*, the value of λ^* is unique and independent of s^*. If $B + \lambda^* M$ is positive definite, s^* is unique.

Suppose the matrix pencil (B, M) has generalized eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and that U is a matrix of associated generalized eigenvectors u_i, $1 \leq i \leq n$, such that $U^{\mathrm{T}}MU = UMU^{\mathrm{T}} = I$. For any scalar λ, let $s(\lambda)$ be the solution to

$$(B + \lambda M)s = -g, \tag{2.3.19}$$

whenever the system is compatible. Zero-finding methods based on Theorem 2.3.3 attempt to find a pair (s, λ) with $\lambda \geq \max(-\lambda_1, 0)$ for the *secular* equation

$$\|s(\lambda)\|_M^{\beta} = \left(\frac{\lambda}{\sigma}\right)^{\beta} \tag{2.3.20}$$

with some appropriately chosen parameter β. The solution of equation (2.3.20) always exits, with the exception of a "hard case" as described in [34]. The hard case may occur when g lies in the orthogonal complement of the eigenspace associated with the smallest generalized eigenvalue λ_1. In this case, as $||s(\lambda)||_M$ has no pole at $\lambda = -\lambda_1$, $||s(\lambda)||_M$ approaches a finite value as $\lambda \to -\lambda_1$. The system (2.3.19) is consistent when $\lambda = \max(-\lambda_1, 0)$, and has the least-length solution $s^\dagger$. However, if $||s^\dagger||_M < \frac{\lambda}{\sigma}$, there is no solution to the secular equation (2.3.20) for $\lambda \geq \max(-\lambda_1, 0)$ (see the right graph in Figure 2.1). In this case, the required solution is $s^\dagger + \tau u_1$, where u_1 is a null vector of $B - \lambda_1 T$ of unit length and the scalar τ is chosen so that $||s^\dagger + \tau u_1||_M = \dfrac{\lambda}{\sigma}$.

To demonstrate the algorithm, the real line is partitioned into three sets:

$$\begin{aligned}
\mathcal{N} &\overset{\text{def}}{=} \{\, \lambda \mid B + \lambda M \text{ is not positive definite.} \,\}, \\
\mathcal{L} &\overset{\text{def}}{=} \{\, \lambda \mid B + \lambda M \text{ is positive definite and } \|s(\lambda)\|_M \geq \frac{\lambda}{\sigma}. \,\}, \\
\mathcal{G} &\overset{\text{def}}{=} \{\, \lambda \mid B + \lambda M \text{ is positive definite and } \|s(\lambda)\|_M < \frac{\lambda}{\sigma}. \,\}.
\end{aligned}$$

and denote $\mathcal{F} = \mathcal{L} \cup \mathcal{G}$ for the set of λ such that $B + \lambda M$ is positive definite. For convenience, define the *secular* function

$$\psi(\lambda; \beta) \overset{\text{def}}{=} \|s(\lambda)\|_M^\beta, \tag{2.3.21}$$

and denote the right hand side of the secular equation (2.3.20) as

$$\tau(\lambda; \beta) \overset{\text{def}}{=} \left(\frac{\lambda}{\sigma}\right)^\beta. \tag{2.3.22}$$

The plots in Figure 2.1 illustrate examples of the function $\psi(\lambda; \beta)$ for a special case of $\beta = 2$, which show the behavior of the function with respect to the value of λ. These plots can be useful to understand how the function behaves in the "easy" case and "hard" case.

The following lemma states the main properties of $\psi(\lambda; \beta)$ for different values of β.

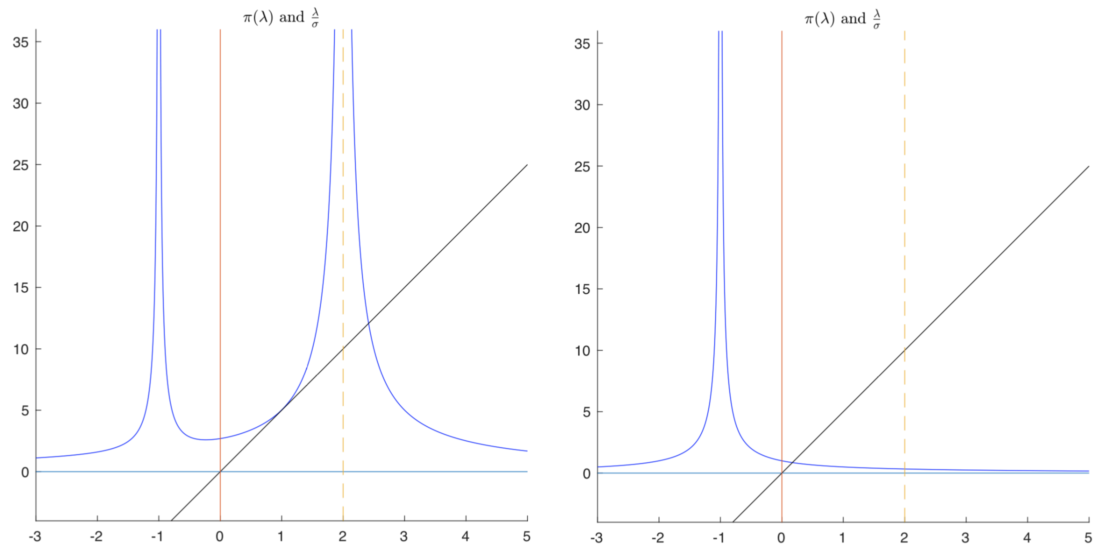

Figure 2.1. Graphs for the problem min $x_1^2 - 2x_2^2 + x_1 + 5x_2 + \dfrac{\sigma}{3}\|x\|^3$ with $\sigma = 0.2$ (the "easy" case, left), and those for min $x_1^2 - 2x_2^2 + x_1 + \dfrac{\sigma}{3}\|x\|^3$ with $\sigma = 0.2$ (the "hard" case, right). Here the x-axis denotes different values of λ, and the y-axis denotes the values of $\pi(\lambda) \overset{\text{def}}{=} \psi(\lambda; 2)$.

Lemma 2.3.2. *Suppose $\psi(\lambda;\beta)$ is defined as in (2.3.21). Then for $\lambda > \max(\lambda_1, 0)$, we have*

(i) $\psi(\lambda;\beta)$ is twice-continuously differentiable;

(ii) $\psi(\lambda;\beta)$ is strictly decreasing to zero and strictly convex when $\beta > 0$;

(iii) $\psi(\lambda;\beta)$ is strictly increasing to infinity and concave when $\beta \in [-1, 0)$.

Proof. As

$$\psi(\lambda;\beta) = \|s(\lambda)\|_M^\beta = \| - (B + \lambda M)^{-1} g\|_M^\beta = \left(\sqrt{\sum_{i=1}^n \frac{(u_i^{\mathrm{T}} M g)^2}{(\lambda + \lambda_i)^2}}\right)^\beta,$$

the results follow from Lemma 2.1 in [10]. $\square$

The uniqueness of the root, say $\lambda^* > \lambda_s$, for the secular equation (2.3.20)

$$\psi(\lambda;\beta) = \tau(\lambda;\beta),$$

is guaranteed by Lemma 2.3.2. In order to solve the secular equation (2.3.20) by safeguarded Newton-like method, the derivatives of $\psi(\lambda;\beta)$ are needed. Let's first consider the derivatives for the special case $\beta = 2$. The derivatives for the general case in which $\beta \neq 2$ can be deduced by chain rule based on the derivatives of $\psi(\lambda;2)$. The derivatives of the function $\psi(\lambda;2)$ with respect to λ can be evaluated recursively using the chain rule as shown in the following result.

Theorem 2.3.4 (Theorem 3 [28]). *Suppose that $H + \lambda M$ is positive definite, and $s(\lambda)$ is the solution to* (2.3.19)*. Let $s^{(0)} \overset{\text{def}}{=} s(\lambda)$ and $\alpha_0 \overset{\text{def}}{=} 1$. Then for $k = 0, 1, \ldots,$ the derivatives of $\psi(\lambda;2)$ satisfy*

$$\psi^{(2k+1)}(\lambda;2) = 2\alpha_k s^{(k)T}(\lambda) M s^{(k+1)}(\lambda),$$
$$\text{and } \psi^{(2k+2)}(\lambda;2) = 2\alpha_{k+1} s^{(k+1)T}(\lambda) M s^{(k+1)}(\lambda)$$

where

$$(H + \lambda M)s^{(k+1)}(\lambda) = -(k+1)Ms^{(k)}(\lambda) \ \text{ and } \ \alpha_{k+1} = \frac{2(2k+3)}{k+1}\alpha_k.$$

For the general case where $\beta \neq 2$, the derivatives of $\psi(\lambda;\beta)$ are expressed in terms of the derivatives of $\psi(\lambda;2)$, and only the derivatives with order less or equal to 3 are considered due to the increasing computational cost for higher order derivatives. As $\psi(\lambda;\beta) = \|s(\lambda)\|_M^\beta = [\psi(\lambda;2)]^{\beta/2}$, by chain rule, the derivatives for $\psi(\lambda;\beta)$ with respect to

λ can be evaluated as

$$\begin{aligned}
\psi^{(1)}(\lambda;\beta) =& \frac{\beta}{2}[\psi(\lambda;2)]^{\beta/2-1}\psi^{(1)}(\lambda;2), \\
\psi^{(2)}(\lambda;\beta) =& \frac{\beta}{2}[\psi(\lambda;2)]^{\beta/2-1}\psi^{(2)}(\lambda;2) \\
&+ \frac{\beta}{2}\left(\frac{\beta}{2}-1\right)[\psi(\lambda;2)]^{\beta/2-2}\left[\psi^{(1)}(\lambda;2)\right]^2, \\
\psi^{(3)}(\lambda;\beta) =& \frac{\beta}{2}[\psi(\lambda;2)]^{\beta/2-1}\psi^{(3)}(\lambda;2) \\
&+ \frac{3\beta}{2}\left(\frac{\beta}{2}-1\right)[\psi(\lambda;2)]^{\beta/2-2}\psi^{(1)}(\lambda;2)\psi^{(2)}(\lambda;2) \\
&+ \frac{\beta}{2}\left(\frac{\beta}{2}-1\right)\left(\frac{\beta}{2}-2\right)[\psi(\lambda;2)]^{\beta/2-3}\left[\psi^{(1)}(\lambda;2)\right]^3.
\end{aligned} \tag{2.3.23}$$

Equipped with the derivatives, the following theorem compares the values of $\psi(\lambda;\beta)$ and its Taylor series approximations. This theorem allows us to determine when the Taylor approximation overestimates or underestimates the function $\psi(\lambda;\beta)$, and guides the choice of β in Newton-like steps in the algorithm towards finding the root.

Theorem 2.3.5 (Theorem 6 [28]). *Suppose that $H + \lambda M$ is positive definite, and $s(\lambda)$ is the solution to* (2.3.19). *Let $\lambda_s \overset{\text{def}}{=} \max(-\lambda_1, 0)$, and $\psi_k(\delta;\beta)$ be the k-th order Taylor-series approximation to $\psi(\lambda+\delta;\beta)$. Then for any $\lambda \geq \lambda_s$, we have the following results.*

(i) for $\beta > 0$, when $\delta > 0$,

$$\psi(\lambda+\delta;\beta) \leq \psi_2(\delta;\beta) \ \text{ and } \ \psi(\lambda+\delta;\beta) \geq \psi_k(\delta;\beta) \ \text{ for } \ k = 1,3; \tag{2.3.24}$$

while when $\lambda_s - \lambda < \delta < 0$,

$$\psi(\lambda+\delta;\beta) \geq \psi_3(\delta;\beta) \geq \psi_2(\delta;\beta) \geq \psi_1(\delta;\beta); \tag{2.3.25}$$

(ii) otherwise, when $\delta > 0$,

$$\begin{aligned}
&\psi(\lambda+\delta;\beta) \le \psi_1(\delta;\beta) \ \textit{for}\ \beta \in [-1,0), \\
&\psi(\lambda+\delta;\beta) \ge \psi_2(\delta;\beta) \ \textit{for}\ \beta \in [-\frac{2}{3},0), \\
&\psi(\lambda+\delta;\beta) \le \psi_3(\delta;\beta) \ \textit{for}\ \beta \in [-\frac{2}{3},0);
\end{aligned} \tag{2.3.26}$$

while when $\lambda_s - \lambda < \delta < 0$,

$$\begin{aligned}
&\psi(\lambda+\delta;\beta) \le \psi_1(\delta;\beta) \ \textit{for}\ \beta \in [-1,0), \\
&\psi(\lambda+\delta;\beta) \le \psi_2(\delta;\beta) \le \psi_1(\delta;\beta) \ \textit{for}\ \beta \in [-\frac{2}{3},0), \\
&\psi(\lambda+\delta;\beta) \le \psi_3(\delta;\beta) \le \psi_2(\delta;\beta) \le \psi_1(\delta;\beta) \ \textit{for}\ \beta \in [-\frac{2}{5},0).
\end{aligned} \tag{2.3.27}$$

The inequalities in (2.3.24)-(2.3.27) are strict if $\lambda \neq \lambda_s$.

At each iterate λ_i, the safeguarded Newton-like method uses a k-th order Taylor approximation of the function $\psi(\lambda, \beta)$, and solves a polynomial

$$\psi_k(\delta;\beta) = \tau(\lambda_i + \delta;\beta), \tag{2.3.28}$$

to obtain a correction δ to λ_i. Consider the case that the current iterate λ_i is in the region $\mathcal{G}$. For positive β, by inequalities (2.3.25) in Theorem 2.3.5, the largest positive roots of the equation (2.3.28) for $k \in \{1,3\}$ will provide underestimations of λ^*, and for $k = 2$ any positive root of (2.3.28) will overestimate λ^*. Meanwhile, for $\beta \in [-1,\infty)$, by inequalities (2.3.27), the least-negative roots of the equation (2.3.28) for $k = 1$ will lead to guaranteed under-estimates of λ^*. When the current iterate λ_i lies in the range $\mathcal{G}$, for positive β, by inequalities (2.3.24) in Theorem 2.3.5, the least negative roots of the equation (2.3.28) for $k \in \{1,2,3\}$ will give estimates to the left of λ^*, while, for $\beta \in [-1,\infty)$, by inequalities (2.3.26), the least-negative roots of the equation (2.3.28) for $k = 1$ will lead to guaranteed under-estimates of λ^*. For a given β and degree k, the best prediction, denoted as $\lambda_k(\beta)$, is obtained by finding the root of (2.3.28), and adding it to the current iterate λ_i. Unlike trust-region subproblem, where the linear Taylor approximation with $\beta = -1$ is the preferred choice [10], it is not clear which values of β and k are optimal for solving the cubic subproblem. The algorithm will select the best $\lambda_k(\beta)$ from a sample of permissible β and k in practice.

The algorithm for solving the cubic subproblem is summarized in Algorithm 2.3. To start with, the algorithm is provided with an initial estimate λ_0, typically obtained from the previous cubic subproblem in the outer iteration. Then a lower bound of the smallest eigenvalue of the Hessian matrix B is estimated from some numerical methods, such as Rayleigh quotient or Gershgorin's theorems [14]. Having a good estimate of the smallest eigenvalue can make the algorithm converge faster, but it doesn't change the final outcome of the convergence. In the line 8, a routine $chol(\cdot)$ is used to compute a Cholesky factorization of the matrix $B + \lambda_i M$ and efficiently determine whether the matrix is symmetric positive definite. If the Cholesky factorization of the matrix $B + \lambda_i M$ fails, indicating that the matrix is not positive definite, the routine $chol(\cdot)$ can determine the index of the pivot position where the factorization failed and return it as *Rtype*. This information can be used to find a direction of negative curvature, and thus an upper bound on the smallest eigenvalue (see [20]). The value of the next iterate λ_{i+1} is then determined by using the lower and upper estimates as in the line 12. If the matrix $B + \lambda_i M$ is found to be positive definite, the solution s of (2.3.19) can be obtained by using the Cholesky factorization computed earlier. If the current solution s satisfies the secular equation (2.3.20) within the desired tolerance, or the gap between lower and upper bound of λ^* is small enough, the algorithm will terminate and the current s will be returned and used as the search step in the outer loop. Note that other relaxed early termination conditions can also be used [8, 34, 21]. If these termination conditions are not satisfied, the value of λ will be updated based on the region that the current iterate λ_i lies in.

The value of λ will be updated using safeguarded Newton-like method, ensuring that each iterate λ_i remains within the region $\mathcal{F}$. Suppose $\lambda_i \in \mathcal{G}$, i.e. $\|s\| > \dfrac{\lambda_i}{\sigma}$. The algorithm will check if a hard case is present by utilizing a routine, $z_{null}(\cdot)$, to compute an approximated null vector z of $B + \lambda_i M$. If this approximated null vector is good enough, satisfying the conditions in the line 19, the solution is constructed as $s + \tau z$ where τ is a properly chosen scalar such that $||s + \tau z||_M = \dfrac{\lambda_i}{\sigma}$. Otherwise, the value of λ will be updated by selecting the maximum value from among $\lambda_1(-1)$, $\lambda_2(2)$, $\lambda_3(2)$, where the derivatives are calculated using the recursive formula given in (2.3.23). Now suppose $\lambda_i \in \mathcal{L}$, i.e. $\|s\| < \dfrac{\lambda_i}{\sigma}$. The value of λ will be updated by choosing the maximum value among $\lambda_1(-1)$, and $\lambda_3(2)$. It is worth noting that once the current iterate λ_i falls into the region $\mathcal{L}$, the subsequent iterates will be guaranteed to remain in the region $\mathcal{L}$ for any $n \geq i$. This is because the update rule for λ will always lead to a underestimation of λ^* as discussed before. However, when $\lambda_i \in \mathcal{G}$, the next iterate may be located within $\mathcal{N}$. In this scenario, a safeguarding strategy is employed to move the next iterate back into the feasible region $\mathcal{F}$ as outlined in the line 36.

Algorithm 2.3. Schematic outline of solving cubic subproblem.

```
1: function CUBIC_SUBPROBLEM
2:     Initialization: Given $\lambda_0$, $0 < \lambda_U$, $0 < \epsilon \ll 1$, $0 \ll \theta < 1$, $i = 0$;
3:     Estimate the lower bound of the smallest eigenvalue of $B$ as $\bar{\lambda}_{min}$;
4:     Set $\lambda_S \leftarrow \max(0, -\bar{\lambda}_{min})$ and $\lambda_0 \leftarrow \max(\lambda_0, \lambda_S)$;
5:     Set $[\lambda_L,\ \lambda_U] \leftarrow [\lambda_S,\ \max(\lambda_0, \lambda_U)]$;
6:     converged $\leftarrow$ false;
7:     while not converged do
8:         Compute Cholesky factorization of $B + \lambda M$: $[R, Rtype] = chol(B + \lambda_i M)$;
9:         if $Rtype > 0$ then
10:            Possibly compute an estimate $\lambda_E \geq \lambda_1$;
11:            Update $\lambda_L \leftarrow \max(\lambda_L, \lambda_E)$;
12:            $\lambda_{i+1} \leftarrow (1 - \theta) * \lambda_L + \theta * \lambda_U$;
13:        else
14:            Compute a vector $s$ by solving the linear systems
                   $$R^{\mathrm{T}} q = -g; R^{\mathrm{T}} s = q.$$
15:            if $\left| \|s\| - \frac{\lambda_i}{\sigma} \right| < \epsilon$ or $\lambda_U - \lambda_L < \epsilon$ then
16:                $s^* \leftarrow s$; converged $\leftarrow$ true;
17:            else if $\lambda_i \in \mathcal{G}$ then
18:                $z \leftarrow z_{null}(B, \lambda_i, M, s, \sigma)$;
19:                if $z^{\mathrm{T}}(B + \lambda_i M)z < \epsilon(2 - \epsilon)(p^{\mathrm{T}}(B + \lambda_i M)p + \lambda_k(\lambda_k/\mu^C)^2)$ then
20:                    Choose a $\tau$ such that $||s(\lambda_i) + \tau z||_M = \dfrac{\lambda_i}{\sigma}$;
21:                    $s^* \leftarrow s + \tau z$; converged $\leftarrow$ true;
22:                else
23:                    $\lambda_S \leftarrow \max(\lambda_S, -z^{\mathrm{T}}(B + \lambda_i M)z)$;
24:                    $[\lambda_L,\ \lambda_U] \leftarrow [\max(\lambda_L, \lambda_S),\ \min(\lambda_U, \lambda_i)]$;
25:                end if
26:            end if
27:            if not converged then
28:                if $\lambda_i \in \mathcal{L}$ then
29:                    $\lambda_N \leftarrow \max\{\lambda_1(-1), \lambda_3(2)\}$
30:                else
31:                    $\lambda_N \leftarrow \max\{\lambda_1(-1), \lambda_2(2), \lambda_3(2)\}$
32:                end if
33:                if $\lambda_N > \lambda_L$ then
34:                    $\lambda_{i+1} \leftarrow \lambda_N$;
35:                else
36:                    $\lambda_{i+1} \leftarrow (1 - \theta) * \lambda_L + \theta * \lambda_U$;
37:                end if
38:            end if
39:        end if
40:    end while
41: end function
```

2.4 Penalty and Barrier Functions

This section provides a brief survey of penalty and barrier functions used in interior methods for solving nonlinear inequality problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad c(x) \geq 0. \tag{2.4.1}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $f : \mathbb{R}^n \mapsto \mathbb{R}$ are twice-continuous differentiable nonlinear functions.

2.4.1 Conventional Barrier Functions

The logarithmic barrier method is a conventional and popular technique for solving problems with inequality constraints. It involves defining a sequence of nondecreasing barrier parameters $\{\, \mu_k \,\}$, with $\mu_k \to 0$, and constructing a logarithmic barrier function $B(x; \mu_k)$ as follows:

$$B(x; \mu_k) = f(x) - \mu_k \sum_{i=1}^{m} \ln c_i(x).$$

The logarithmic terms are well defined only at points x for which $c(x) > 0$, and becomes unbounded as x approaches points where the constraint is satisfied. The conventional barrier method aims to minimize $B(x; \mu_k)$ subject to the constraint that $c(x) > 0$, and the unconstrained local minimizer $x(\mu_k)$ of $B(x; \mu_k)$ defines a continuously differentiable path known as the barrier trajectory or central path. The goal is to follow this trajectory towards a local solution of the problem (2.4.1). It is important to note that the logarithmic barrier method requires a feasible starting point, and that it converges to a solution under certain updating strategies of the sequence of barrier parameters.

Directly minimizing $B(x; \mu)$ is not recommended because of poor convergence as $\mu \to 0$, as discussed by Forsgren, Gill, and Wright [19]. This is due to the fact that the unconstrained minimizer $x(\mu)$ is a poor estimate of the unconstrained minimizer $x(\bar{\mu})$ when the barrier parameter is decreased from μ to $\bar{\mu}$. As a result, a full Newton step cannot be taken immediately after the barrier parameter is reduced. This inefficiency makes the classical logarithmic barrier method unsuitable for practical use.

2.4.2 Modified Barrier Functions

Modified barrier methods [12, 24, 37] are a class of optimization algorithms that aim to overcome the difficulties associated with minimizing the logarithmic barrier function $B(x;\mu)$ as $\mu \to 0$ in the conventional barrier method. These methods define a sequence of unconstrained problems in which the value of μ is kept bounded away from zero. By doing so, they avoid the problem of dealing with an ill-conditioned Hessian, and can achieve better convergence properties.

Modified barrier methods take advantage of the equivalence between the original inequality constraints $c_i(x) \geq 0$ and the modified inequality constraints $\mu \ln(1 + c_i(x)/\mu) \geq 0$ for fixed positive μ. A KKT point for the original problem (2.4.1) is also a KKT point for the modified problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad \mu \ln(1 + c_i(x)/\mu) \geq 0, \; i = 1, 2, \dots, m. \tag{2.4.2}$$

The modified barrier function $\mathcal{M}(x, \lambda)$ is defined as the conventional Lagrangian function for the modified problem (2.4.2). It incorporates the logarithmic barrier terms, but with the modified constraints $\mu \ln(1 + c_i(x)/\mu) \geq 0$ instead of $c_i(x) > 0$, which is written as

$$\mathcal{M}(x;\lambda) = f(x) - \mu \sum_{i=1}^{m} \lambda_i \ln(1 + c_i(x)/\mu). \tag{2.4.3}$$

The modified barrier function has an important property: if a KKT point x^* has an acceptable multiplier λ^* in $\mathcal{Y}(x^*)$, then there exists a fixed μ^* such that for any μ, the corresponding x^* is a local minimum of $\mathcal{M}(x;\lambda^*)$. This suggests that $\nabla\mathcal{M}(x^*;\lambda^*) = 0$ and $\nabla^2\mathcal{M}(x^*;\lambda^*) = 0$ is positive semidefinite. As a result, if an optimal multiplier is known, finding x^* can be accomplished with a single unconstrained minimization.

In practical applications, both the optimal multiplier vector and an upper bound on μ are typically unknown in advance. Therefore, a series of problems must be solved, where each problem defines the merit function (2.4.3) using estimates of λ^* and μ^*. The multiplier estimate is then updated after each subproblem, and the barrier parameter is decreased if $\nabla^2\mathcal{M}(x;\lambda)$ is not positively definite enough. To find further information, please refer to the following sources [24, 35, 37].

Primal-dual methods have become the most popular class of interior-point methods in recent years. However, if the merit function (2.4.3) is used as the merit function, these methods need a separate safeguarded measurement for the dual variables after the primal step since (2.4.3) does not involve any dual variables. However, since the merit function (2.4.3) does not involve the dual variables, these methods require a separate safeguarded measurement to the dual variables after the primal step, if (2.4.3) is used as the merit function. Furthermore, when the problem is nonconvex, and the current primal-dual iterate is far from the trajectory, there is no assurance that a solution exists for the primal-dual system or condensed system.

An alternative approach is based on a merit function that incorporates both primal and dual variables [18], which is defined as

$$\mathcal{M}^{\mu}(x,y) = f(x) - \mu\sum_{i=1}^{m}\ln c_i(x) - \sum_{i=1}^{m}\left\{\ln\left(\frac{y_i c_i(x)}{\mu}\right) + \left(1 - \frac{y_i c_i(x)}{\mu}\right)\right\}. \tag{2.4.4}$$

The function (2.4.4) can be interpreted as the classical barrier function $B(x;\mu)$ augmented by a weighted proximity term that measures the distance of (x,y) to the trajectory $(x(\mu),y(\mu))$. The key property of this merit function is that it is minimized with respect to both x and y at any point $(x(\mu),y(\mu))$ on the trajectory of minimizers, and a decrease in the function (2.4.4) can be used to encourage progress toward a minimizer of $B(x;\mu)$.

Despite the advantages of the merit function (2.4.4), there are still some limitations.

For example, it can be challenging to handle equality constraints within this scheme, and finding a strictly feasible point can also be difficult in practice. Additionally, the merit function approach still requires the barrier parameter μ to converge to zero, which means that some of the numerical benefits of barrier methods are lost.

2.4.3 Primal-Dual Shifted Penalty-Barrier Functions

Gill, Kungurtsev, and Robinson [22] recently introduced a shifted primal-dual penalty-barrier function that serves as a merit function for a primal-dual path-following method. This method addresses issues with equality constraints and the need for a strictly feasible point, which were discussed in the previous section. We adopt this shifted primal-dual penalty-barrier function as a merit function used in the primal-dual cubic regularization methods described in Chapter 4.

The shifted primal-dual problem associated with problem (2.4.1) is obtained by including the constraints $c(x) - s = 0$ with the objective using a shifted primal-dual augmented Lagrangian term, and using a shifted primal-dual penalty-barrier term for the simple bounds $s \geq 0$. This gives the problem

$$\underset{x,s,y,w}{\text{minimize}}\ \mathcal{M}(x,s,y,w\,;\mu^{P},\mu^{B},y^{E},w^{E}) \quad \text{subject to} \quad s+\mu^{B}e>0, \quad w>0, \tag{2.4.5}$$

where $\mathcal{M}(x,s,y,w\,;\mu^{P},\mu^{B},y^{E},w^{E})$ is the shifted primal-dual penalty-barrier function

$$\begin{aligned}\mathcal{M}(x,s,w,y;w^{E},y^{E},\mu^{P},\mu^{B}) &= f(x) - \big(c(x)-s\big)^{\mathrm{T}} y^{E} \\ &\quad + \frac{1}{2\mu^{P}}\|c(x)-s\|^{2} + \frac{1}{2\mu^{P}}\|c(x)-s+\mu^{P}(y-y^{E})\|^{2} \\ &\quad - \sum_{i=1}^{m}\mu^{B}w_i^{E}\ln\big(s_i+\mu^{B}\big) - \sum_{i=1}^{m}\mu^{B}w_i^{E}\ln\big(w_i(s_i+\mu^{B})\big) + \sum_{i=1}^{m} w_i(s_i+\mu^{B}).\end{aligned} \tag{2.4.6}$$

which is well defined for all w and s such that $w>0$ and $s+\mu^{B}e>0$.

The search directions generated by a specific approximate Newton method for minimizing the merit function (2.4.6) can be shown to be identical to those of a variant of the conventional path-following method in which the perturbation of the complementarity condition does not need to go to zero. By updating the parameters appropriately, the stationary points of the merit function (2.4.6) have properties that can be used in the formulation of a globally convergent algorithm for the nonlinear programming problem (2.4.1). For a more comprehensive explanation and analysis of the shifted primal-dual penalty-barrier function and its associated algorithm, please refer to the work of Gill, Kungurtsev and Robinson [22].

Chapter 3

Cubic Regularization Algorithms for Unconstrained Optimization

This chapter addresses the application of the cubic regularization method to unconstrained optimization. Consider the unconstrained nonlinear programming problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \tag{UC}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth function. The primary objective of this study is to improve the numerical efficiency of the ARC algorithm for solving problem (UC) while preserving the convergence properties.

Two improvements to the ARC algorithm are proposed. The first involves the use of a line search with the cubic trial step. The second emphasizes the employment of Newton's step when appropriate. In Section 3.1, we formulate and analyse a cubic regularization algorithm in conjunction with a nonmonotone line search that employs the strong Wolfe conditions, and investigate the convergence properties of the proposed algorithm. Section 3.2 explores two hybrid methods that integrate cubic regularization and Newton's method, and examines their respective properties.

3.1 Cubic Regularization Methods with Line Search Techniques

3.1.1 Introduction and Motivations

This section introduces and examines a novel algorithm that combines cubic regularization methods and line-search techniques. One of the well-known limitations of cubic regularization frameworks is that solving the cubic regularized subproblem can involve solving one or more linear systems or may execute an iterative process that incurs significant computational cost to the method. In contrast, a line-search method, despite possibly requiring a higher number of iterations and function evaluations to determine a minimum of the objective function f, generally achieves greater efficiency per iteration. Motivated by these observations, when evaluating the objective function is not computationally intensive, performing a line search on the objective function along the direction of the rejected trial step can eliminate the need to repeatedly solve the cubic subproblem when the sufficient decrease condition is not satisfied. This approach leads to a reduced number of iterations and ultimately reduces the overall computational cost.

Bianconcini and Sciandrone [4] proposed a modification to the ARC algorithm that enhances its computational efficiency while preserving global convergence properties. Their approach is based on a suitably combination of the Armijo-Goldstein line search and the ARC method. The concept behind this approach is to leverage the computational effort spent on minimizing the cubic model to compute the search direction and exploit the potential desirable properties of this direction. Recently, Dehghan, Heydari and Hosseini [15] presented two modified versions of the ARC method for the unconstrained optimization problems, where the trial step is accepted by a nonmonotone Armijo-type line search technique. The nonmonotone methods stand out for not enforcing a strict monotonicity in the objective function values at successive iterations. Some studies have shown that the use of nonmonotone techniques can improve the chances of finding the global optimum and accelerate the rate of convergence [30, 39]. These methods typically exhibit advantageous numerical results, particularly when applied to highly nonlinear problems.

Although the Armijo-type line search on the cubic trial is a straightforward and easy-to-implement method, an unsatisfactory feature of it is that the step size can only be decreased during the line search process. The Armijo-type line search can result in a step size that is too small, which can slow down convergence, especially in cases where the function being optimized is not well-behaved or has flat regions. The proposed algorithm in this section addresses this limitation by implementing a nonmonotone line search technique with the strong Wolfe conditions on the cubic step. The Wolfe line search is a more sophisticated line search method that provides both a sufficient decrease condition and a curvature condition. It can result in a larger step size and converge faster, especially for nonsmooth or ill-conditioned objective functions.

In Section 3.1.2, the design and implementation of the NMARC algorithm are presented. The theoretical analysis of the convergence of the NMARC algorithm is provided in Section 3.1.3. The numerical results of the NMARC algorithm on CUTEst unconstrained test problems and ablation studies are reported in Section 5.2.

3.1.2 A Nonmonotone ARC algorithm

This section commences with the introduction of a nonmonotone approach incorporating the Wolfe conditions, followed by a description of our nonmonotonic ARC line-search framework.

As suggested by Zhang and Hager [39], the strongest nonmonotone strategies yield the best convergence results when the iterations are far from the optimal solution, and weaker strategies perform better when closer to the minimizer. The nonmonotone term is defined as

$$Q_{k+1} = \beta_k Q_k + 1, \quad C_{k+1} = \frac{\beta_k Q_k C_k + f_{k+1}}{Q_{k+1}}, \tag{3.1.1}$$

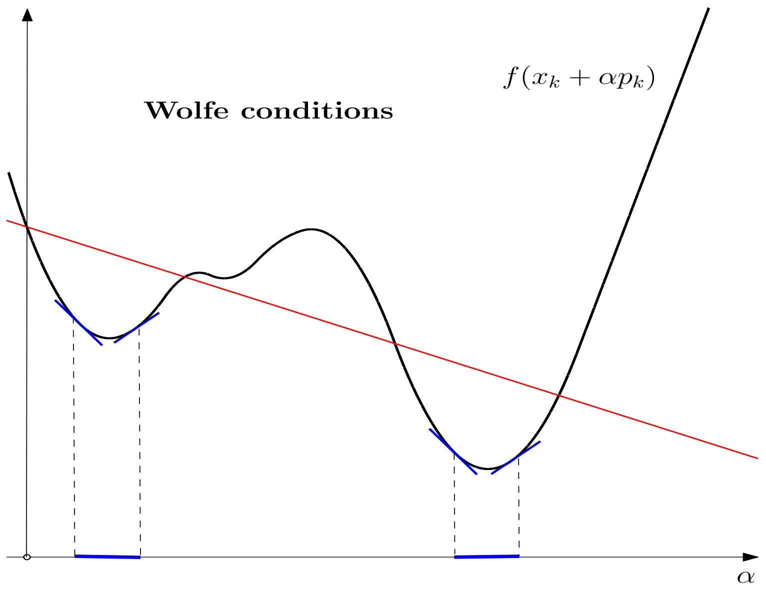

Figure 3.1. The red line denotes a linear function derived from the Armijo condition $l_k(x) = f_k + \eta_s \alpha g_k^{\mathrm{T}} p_k$, which ensures a sufficient reduction in the objective function value. The blue line segments represent a set of α values that satisfy the strong Wolfe conditions.

where $Q_0 = f_0$, $\beta_k \in [\beta_{\min}, \beta_{\max}]$, $\beta_{\min} \in [0, 1)$, and $\beta_{\max} \in [\beta_{\min}, 1)$. It can be seen that the value of C_{k+1} is a weighted average of C_k and f_{k+1}, where the value of C_k at each iteration is a convex combination of the function values at all previous iterations, f_0, f_1, ..., f_k. The value of β_k plays a crucial role in determining the level of nonmonotonicity in the algorithm. When the value of β_k is close to 1, it results in a stronger nonmonotone strategy that puts equal weight on all the previous function values. On the other hand, a value closer to 0 results in a weaker nonmonotone strategy which approximates a traditional monotone line search by giving more importance to the recent function values. By adaptive selecting β_k, the influence of C_k can be increased when far from the optimal solution and decreased when closer to it.

Drawing inspiration from the biased Wolfe trust-region algorithm introduced by Gertz [33], the cubic regularization step can be augmented with a line search that locates a point that adheres to the Wolfe conditions. Figure 3.1 depicts an example of a set of points meeting the (monotone) strong Wolfe conditions. A rough approximation of an exact line search can be achieved by requiring the magnitude of $g(x_k + \alpha p_k)^{\mathrm{T}} p_k$ to be sufficiently reduced compared to $g_k^{\mathrm{T}} p_k$. Let η_w satisfy $0 \le \eta_w < 1$. The first Wolfe condition on α can be written as

$$|g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k| \le \eta_w |g_k^{\mathrm{T}} p_k|. \tag{3.1.2}$$

This condition prevents the step length α_k from being not too small. The second (non-monotone) Wolfe condition serves the purpose of ensuring that the trial step is making sufficient progress towards a decrease in function value, similar to the Armijo condition. It is usually stated as:

$$f(x_k + \alpha_k p_k) \le C_k + \eta_s \alpha_k g_k^{\mathrm{T}} p_k, \tag{3.1.3}$$

where η_s is a constant satisfying $0 < \eta_s < \frac{1}{2}$. This criterion ensures that the trial step is not too conservative, and that the function value is reducing by a sufficient amount. Note that if η_{k-1} is set to 0 in (3.1.1), then $C_k = f_k$ and the condition (3.1.3) is the usual Armijo condition. Together, theses two criteria (3.1.2) and (3.1.3) characterize a set of acceptable step lengths that satisfy the strong Wolfe conditions.

Algorithm 3.1. Schematic outline of NMARC.

```
1: function Nonmontone_ARC
2:     Initialization: Given $x_0$, $0 < \eta_1 \le \eta_2 < 1$, $1 < \gamma_1 \le \gamma_2$, $\sigma_0 > 0$.
3:     $k = 0$; $C_0 = f_0$, $Q_0 = 1$.
4:     while not converged do
5:         Compute a step $p_k$ for which
               $$m_k(p_k) \le m_k(p_k^C), \qquad (3.1.4)$$
           where the Cauchy step
               $$p_k^C = -\alpha_k^C g_k \text{ and } \alpha_k^C = \operatorname*{argmin}_{\alpha \in \mathbb{R}_+} m_k(-\alpha_k g_k). \qquad (3.1.5)$$
6:         Compute
               $$\rho_k = \frac{C_k - f(x_k + p_k)}{f_k - m_k(p_k)}.$$
7:         if $\rho_k \ge \eta_1$ then
8:             Set $x_{k+1} = x_k + p_k$.
9:             Set $\alpha_k = \begin{cases} 2 & \text{if } \rho_k \ge \eta_2, \\ 1 & \text{otherwise.} \end{cases}$
10:        else
11:            Set $x_{k+1} = x_k + \alpha_k p_k$ where $\alpha_k$ satisfies Wolfe conditions:
               $$\begin{aligned} f(x_k + \alpha_k p_k) &\le C_k + \eta_s \alpha_k g_k^{\mathrm{T}} p_k, & \eta_s &\in (0, \tfrac{1}{2}), \\ |g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k| &\le \eta_w |g_k^{\mathrm{T}} p_k|, & \eta_w &\in (\eta_s, 1). \end{aligned} \qquad (3.1.6)$$
12:        end if
13:        Cost update: Choose $\beta_k \in [0, 1)$, and set
               $$Q_{k+1} = \beta_k Q_k + 1, \quad C_{k+1} = \frac{\beta_k Q_k C_k + f_{k+1}}{Q_{k+1}}.$$
14:        Update
               $$\sigma_{k+1} \in \begin{cases} (0, \sigma_k) & \text{if } \alpha_k > 1, \\ [\sigma_k, \gamma_1 \sigma_k) & \text{if } \alpha_k = 1, \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k) & \text{otherwise.} \end{cases}$$
15:        $k = k + 1$.
16:    end while
17: end function
```

The proposed nonmonotone adaptive cubic regularization (NMARC) method is summarized in Algorithm 3.1. The algorithm starts by solving the cubic subproblem

$$m_k(s) \overset{\text{def}}{=} f_k + g_k^{\mathrm{T}} s + \tfrac{1}{2} s^{\mathrm{T}} B_k s + \frac{1}{3}\sigma_k \|s\|^3$$

to compute the trial step p_k. Then the ratio ρ_k of the actual reduction to the predicted reduction in objective is computed as

$$\rho_k = \frac{C_k - f(x_k + p_k)}{f_k - m_k(p_k)}. \tag{3.1.7}$$

If the trial step p_k is accepted by nonmonotone criteria, the step is considered to be successful and the next iterate is set as $x_{k+1} = x_k + p_k$. Otherwise, the method performs a nonmonotone line search in the direction of the rejected trial step p_k instead of resolving the cubic subproblem again. The step size α_k that satisfies the nonmonotone Wolfe conditions (3.1.6) always exists, as demonstrated in Lemma 3.1.3 in the following section, which implies that the line search update is well-defined. Then, nonmonotone term is updated using (3.1.1). Additionally, the weight σ_k will be updated properly according to the step size α_k. If the iteration is successful or α_k is not reduced during the line search process (i.e., $\alpha_k \geq 1$), then σ_k may remain unchanged or be decreased. Conversely, if the step length is reduced, the weight σ_k will be increased prior to the next iteration, with the aim of reducing the step size.

3.1.3 Theoretical Discussion

In this section, the first-order convergence results of Algorithm 3.1 will be discussed. Under certain mild assumptions, it can be shown that the sequence $\{x_k\}$ produced by Algorithm 3.1 converges towards a solution with $\|g_k\| \to 0$ as $k \to \infty$ Similarly to the ARC algorithm, Algorithm 3.1 only requires an approximate solution that is no worse than the Cauchy step for convergence.

To prove the convergence of Algorithm 3.1, we first present a series of supporting lemmas. The following lemma, which was previously proved in [8], provides useful properties of the approximate solution of the cubic subproblem satisfying (3.1.4).

Lemma 3.1.1 (Properties of a Cauchy step)**.** *Suppose that p_k^C is the Cauchy step, i.e. p_k^C is the solution of* (3.1.5)*, and the step p_k satisfies* (3.1.4)*. Then for $k \geq 0$, we have that*

$$f_k - m_k(p_k) \geq f_k - m_k(p_k^C) \geq \frac{\|g_k\|}{6\sqrt{2}} \min\left[\frac{\|g_k\|}{1+\|B_k\|}, \frac{1}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}\right]. \tag{3.1.8}$$

The next lemma shows a upper bound for the size of the approximate solution of the cubic subproblem in terms of the regularization parameter σ_k. It was also proved previously in [8].

Lemma 3.1.2. *Assume $\|B_k\| \leq \kappa_B$ for all $k \geq 0$ and for some constant $\kappa_B > 0$. Suppose that $\mathcal{I}$ is an infinite index set such that $\alpha_k > 0$ and $\|g_k\| \geq \epsilon$, for all $k \in \mathcal{I}$ and sine $\epsilon > 0$, and $\sqrt{\|g_k\|/\sigma_k} \to 0$, as $k \to \infty$, $k \in \mathcal{I}$. Then*

$$\|p_k\| \leq 3\sqrt{\frac{\|g_k\|}{\sigma_k}} \quad \textit{for all } k \in \mathcal{I} \textit{ sufficiently large.} \tag{3.1.9}$$

Additionally, if $x_k \to x^*$, as $k \in \mathcal{I}$, $k \to \infty$ for some $x^* \in \mathbb{R}^n$, then each iteration $k \in \mathcal{I}$ that is sufficiently large is very successful, and

$$\sigma_{k+1} \leq \sigma_k, \quad \textit{for all } k \in \mathcal{I} \textit{ sufficiently large.} \tag{3.1.10}$$

To ensure that the nonmonotone line search update with strong Wolfe conditions is well-defined, sufficient decrease conditions will be suitable for practical algorithms only when points exist that satisfy them, i.e., it is necessary to guarantee that the set of acceptable step lengths is not empty. The next result shows that, when f is twice-continuously differentiable, an interval of positive steps satisfying the strong Wolfe conditions exists as long as $\eta_w \geq \eta_s$.

Lemma 3.1.3. *Consider a line-search algorithm with initial point x_0, such that the level set $\mathcal{L}(f_0)$ is bounded, and assume that p_k an approximate solution of the cubic subproblem $m_k(s)$ satisfying* (3.1.4) *for all $k \geq 0$. Then at every iteration k there exists $\alpha_l > 0$ and an interval (α_l, α_u) such that the strong Wolfe conditions,*

$$|g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k| \leq \eta_w |g_k^{\mathrm{T}} p_k| \quad \textit{and} \quad f(x_k + \alpha_k p_k) \leq C_k + \eta_s \alpha_k g_k^{\mathrm{T}} p_k,$$

are satisfied for every $\alpha \in (\alpha_l, \alpha_u)$.

Proof. This proof is divided into two parts. In the first part, we will prove that there exists a positive scalar ξ that satisfies the nonmonotone Armijo condition (3.1.3) and also $g(x + \xi p_k)^{\mathrm{T}} p_k = \eta_s g(x_k)^{\mathrm{T}} p_k$. In the second part, we will demonstrate that such a ξ also satisfies the condition in (3.1.2).

For simplicity, let $M(\alpha)$ denote the univariate function $M(\alpha) = f(x_k + \alpha p_k)$, with $M(0) = f_k$, $M'(0) = g_k^{\mathrm{T}} p_k$. The nonmonotone strong Wolfe conditions (3.1.2) and (3.1.3) can be written as

$$|M'(\alpha)| \leq \eta_w |M'(0)| \quad \text{and} \quad M(\alpha) \leq C_k + \alpha \eta_s M'(0).$$

Let $s(\alpha)$ denote the univariate function

$$s(\alpha) = f(x_k + \alpha_k p_k) - C_k - \alpha \eta_s g_k^{\mathrm{T}} p_k = M(\alpha) - C_k - \alpha \eta_s M'(0).$$

Taking the derivative of $s(\alpha)$ with respect to α gives

$$s'(\alpha) = M'(\alpha) - \alpha \eta_s M'(0),$$

From Lemma 3.1.1, it can be shown that p_k is a descent direction, so that $M'(0) < 0$.

Together with $\eta_s < 1$, we have

$$s'(0) = M'(0) - \eta_s M'(0) = (1 - \eta_s)M'(0) < 0.$$

The nonzero derivative theorem implies the existence of a positive scalar $\hat{\alpha}$ such that $s(\alpha) < 0$ for all $\alpha \in (0, \hat{\alpha})$. As a result, there exists a scalar $\xi_1 \in (0, \hat{\alpha})$ such that $s(\xi_1) < 0$.

Since the level set is bounded, it is compact and $M(\alpha)$ is bounded below by some constant L, i.e., $M(\alpha) \geq L$ for all $\alpha \in [0, \infty)$. Since $C_k + \alpha\eta_s M'(0) \to -\infty$ as $\alpha \to +\infty$, there must exist a positive ξ_2 such that

$$C_k + \xi_2\eta_s M'(0) = L,$$

and we have

$$s(\xi_2) = M(\alpha) - C_k - \xi_2\eta_s M'(0) \geq L - C_k - \xi_2\eta_s M'(0) = 0.$$

Let $\bar{\alpha}$ denote the smallest positive root of $s(\alpha) = 0$. Since $s(0) = 0$, $s(\bar{\alpha}) = 0$, and $s(\alpha) < 0$ for all $\alpha \in (0, \bar{\alpha})$, by mean value theorem, there must exist an $\xi \in (0, \bar{\alpha})$ such that $s'(\xi) = 0$ and $s(\xi) < 0$, or, equivalently,

$$M'(\xi) = \eta_s M'(0) \quad \text{and} \quad M(\xi) < C_k + \xi\eta_s M'(0).$$

Moreover, we know that $M'(0) < 0$, so $M'(\xi) < 0$. Additionally, the inequality $\eta_s \leq \eta_w$ implies that

$$M'(\xi) \geq \eta_s M'(0) \geq -\eta_w M'(0).$$

These inequalities lead to

$$\eta_w M'(0) \leq M'(\xi) < 0 \leq -\eta_w M'(0),$$

which implies that ξ satisfies $|M'(\xi)| \leq \eta_w |M'(0)|$.

We have demonstrated the existence of a positive scalar ξ that satisfies both the nonmonotone Armijo condition (3.1.3) and the curvature condition (3.1.2), thus indicating that the set of points satisfying the nonmonotone strong Wolfe conditions is non-empty. □

The next auxiliary lemma shows the sequence $\{C_k\}$ generated by Algorithm 3.1 is monotonically decreasing.

Lemma 3.1.4. *Let $\{x_k\}$ be the sequence generated by Algorithm 3.1. Then for any $k \geq 0$, we have*

$$f_{k+1} \leq C_{k+1} \leq C_k. \tag{3.1.11}$$

Proof. In each iteration of the algorithm, there are two possibilities: either the search direction p_k is accepted when $\rho_k \geq \eta_1$, or a line search is initiated. If $\rho_k \geq \eta_1$, following from (3.1.7) and Lemma 3.1.1, it can be deduced that

$$C_k - f_{k+1} \geq \eta_1 \left(f_k - m_k(p_k)\right) \geq 0.$$

If, on the other hand, by (3.1.4) and (3.1.3), we have

$$f_{k+1} \leq C_k + \eta_s \alpha_k g_k^{\mathrm{T}} p_k \leq C_k.$$

Thus, through the updating rule of C_k as defined in (3.1.1), it follows that result (3.1.11) holds. □

For simplicity, we denote the index set of all successful iterations of the NMARC algorithm by

$$\mathcal{S} = \{k > 0 : C_k - f(x_k + p_k) \leq \eta_1 \left(f_k - m_k(p_k)\right)\}.$$

Theorem 3.1.1. *Assume f is continuously differentiable over $\mathbb{R}^n$ and $\|B_k\| \leq \kappa_B$ for all*

$k \geq 0$ and for some constant $\kappa_B > 0$. If $\{f_k\}$ is bounded below, then

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.1.12}$$

Proof. We will prove this theorem by contradiction. Assume that the result in equation (3.1.12) is not true, meaning that there exists some $\epsilon > 0$ such that

$$|g_k| \geq \epsilon, \text{ for all } k \geq 0. \tag{3.1.13}$$

To show the contradiction, we will demonstrate that the following equation holds,

$$\sum_{k=1}^{\infty} \sqrt{\frac{\|g_k\|}{\sigma_k}} < +\infty. \tag{3.1.14}$$

If $k \in \mathcal{S}$, by the definition of ρ given in (3.1.7) and using Lemma 3.1.1, it follows that

$$C_k - f_{k+1} \geq \eta_1 \left(f_k - m_k(p_k)\right) \geq \eta_1 \frac{\|g_k\|}{6\sqrt{2}} \min\left\{\frac{\epsilon}{1+\kappa_B}, \tfrac{1}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}\right\}. \tag{3.1.15}$$

On the other hand, if $k \notin \mathcal{S}$, it can be proved from Lemma 3.1.1, Lemma 3.1.3, and the construction of Algorithm 3.1 that

$$-g_k^{\mathrm{T}} p_k \geq f_k - m_k(p_k) \geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{\frac{\epsilon}{1+\kappa_B}, \tfrac{1}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}\right\},$$

and hence

$$f_{k+1} \leq C_k + \eta_s \alpha_l g_k^{\mathrm{T}} p_k \leq C_k - \eta_s \alpha_l \frac{\|g_k\|}{6\sqrt{2}} \min\left\{\frac{\epsilon}{1+\kappa_B}, \tfrac{1}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}\right\}. \tag{3.1.16}$$

By combining (3.1.15) and (3.1.16), it can be deduced that

$$f_{k+1} \leq C_k - \frac{\|g_k\|}{6\sqrt{2}} \min\left\{\frac{\eta_1 \epsilon}{1+\kappa_B}, \frac{\eta_1}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}, \frac{\eta_s \alpha_l \epsilon}{1+\kappa_B}, \frac{\eta_s \alpha_l}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}\right\}.$$

Together with the definition of C_k in (3.1.1) and Lemma 3.1.1, we can write

$$C_k - C_{k+1} \geq \frac{\epsilon}{6\sqrt{2}Q_{k+1}} \min\left\{\frac{\eta_1 \epsilon}{1+\kappa_B}, \frac{\eta_1}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}, \frac{\eta_s \alpha_l \epsilon}{1+\kappa_B}, \frac{\eta_s \alpha_l}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}\right\}.$$

Since $\beta_{\max} < 1$ and $Q_{k+1} \leq 1 + \sum_{j=0}^{k} \beta_{\max}^{j+1} \leq \dfrac{1}{1-\beta_{\max}}$, it follows that

$$C_k - C_{k+1} \geq \frac{\epsilon}{6\sqrt{2}(1-\beta_{\max})} \min\left\{\frac{\eta_1 \epsilon}{1+\kappa_B}, \frac{\eta_1}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}, \frac{\eta_s \alpha_l \epsilon}{1+\kappa_B}, \frac{\eta_s \alpha_l}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}\right\}. \quad (3.1.17)$$

Since $\{f_k\}$ is assumed to be bounded below and by utilizing Lemma 3.1.4 which states that $f_k \leq C_k \leq C_{k-1}$, $\{C_k\}$ is monotonically decreasing and bounded below, and and thus it is convergent. This implies that

$$\sum_{k=0}^{\infty}(C_k - C_{k+1}) \leq C_0 < +\infty,$$

Therefore, it follows from (3.1.15) and (3.1.17) that (3.1.14) holds, and

$$\lim_{k\to\infty} \sqrt{\frac{\|g_k\|}{\sigma_k}} = 0. \quad (3.1.18)$$

Hence, as a result of Lemma 3.1.2 and the construction of Algorithm 3.1, we can conclude that for sufficiently large $l \geq 0$ and $r \geq 0$, we have

$$\begin{aligned}\|x_{l+r} - x_l\| &\leq \sum_{k=l}^{l+r-1} \|x_{k+1} - x_k\| \\ &= \sum_{k=l}^{l+r-1} \alpha_k \|p_k\| \\ &\leq 3\alpha_u \sum_{k=l}^{l+r-1} \sqrt{\frac{\|g_k\|}{\sigma_k}},\end{aligned}$$

which implies that $\{x_k\}$ generated by Algorithm 3.1 is a Cauchy convergent sequence, meaning there exists a point $x^* \in \mathbb{R}^n$ such that $x_k \to x^*$ as $k \to \infty$. By Lemma 3.1.2, we know that $\{\sigma_k\}$ is a bounded sequence, i.e., there exists a constant $\hat{\sigma} > 0$ such that $\sigma_k \leq \hat{\sigma}$. This leads to

$$\sqrt{\frac{\|g_k\|}{\sigma_k}} \geq \sqrt{\frac{\epsilon}{\hat{\sigma}}} > 0,$$

which contradicts with (3.1.18). The proof of the theorem is completed. $\square$

The following corollary demonstrates that the implementation of the line-search technique in Algorithm 3.1 will not compromise the first-order convergence properties of the cubic regularization algorithm. Under the assumptions stated in Theorem 3.1.1 and with the additional requirement of uniform continuity of the gradient g on the iterates $\{x_k\}$, it can be shown that Algorithm 3.1 will either reach a termination point where the convergence criteria is satisfied, or $\lim_{k\to\infty} \|g_k\| = 0$.

Corollary 3.1.1. *Assume $\|B_k\| \leq \kappa_B$ for all $k \geq 0$ and for some constant $\kappa_B > 0$. If $\{f_k\}$ is bounded below and f and g are uniformly continuous on the sequence $\{x_k\}$, then*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.1.19}$$

Proof. We shall prove this theorem by contradiction. To this end, suppose that the conclusion in equation (3.1.19) does not hold, i.e., that there exists an infinite subsequence

$\{t_i\}$ and a constant $\epsilon > 0$ such that

$$\|g_{t_i}\| \geq 2\epsilon, \quad \text{for all } i > 0. \tag{3.1.20}$$

This implies that the gradient of the objective function is not converging to zero along this subsequence. For all $i > 0$, let l_i denote the first iteration after t_i such that $\|g_{l_i}\| \leq \epsilon$, and define the set $\mathcal{K} = \{k \geq 0 : t_i \leq k < l_i\}$. It follows that

$$\|g_k\| \geq \epsilon \text{ for all } t_i \leq k < l_i, \text{ and } \|g_{l_i}\| < \epsilon. \tag{3.1.21}$$

for all $i > 0$. It is important to note that $\mathcal{K}$ is an infinite set. Subsequently, following the same reasoning used in the proof of Theorem 3.1.1, we would obtain that

$$\frac{36\sqrt{2}}{\epsilon \min\{\eta_1, \eta_s \alpha_l\}} (C_{t_i} - C_{l_i}) \geq \sum_{k=t_i}^{l_i - 1} \|x_{k+1} - x_k\| \geq \|x_{l_i} - x_{t_i}\|, \tag{3.1.22}$$

Additionally, the uniform continuity assumption of g on the sequence of iterates $\{x_k\}$ implies that

$$\|g_{l_i} - g_{t_i}\| \to 0 \text{ as } i \to \infty. \tag{3.1.23}$$

However, the inequalities (3.1.21) and (3.1.21) lead to

$$\|g_{l_i} - g_{l_i}\| \geq \|g_{t_i}\| - \|g_{l_i}\| \geq \epsilon,$$

for all $i \geq 0$, in contradiction with the result (3.1.23). As such, the proof of the theorem is now complete. $\square$

3.2 Hybrid Approaches of Cubic Regularization and Newton's Method

3.2.1 Introduction and Motivations

The purpose of combining the cubic regularization method with Newton's method is to mitigate the computational cost associated with minimizing the cubic subproblem. In general, the Newton method is a popular choice for its fast rate of convergence, however, it can become impractical in certain situations where the Hessian is not positive definite or is nearly singular. The cubic regularization method is particularly useful when the objective function is not well-behaved, such as when it is non-convex, has many local minima, or is poorly conditioned. In such cases, regularization can improve the robustness and efficiency of the optimization process and prevent numerical instability by adding a regularization term to the objective function that encourages the iterates to converge to a local minimizer. The cubic regularization method works by adding a cubic regularization term to overcome the non-convexity. This term serves to ensure the overestimation property and the regularization parameter implicitly controls the step size.

As discussed in Section 2.3, the regularization parameter for the cubic term is initialized at the start of the algorithm and subsequently updated at each iteration using a scheme that is based on ensuring sufficient descent. This approach can significantly influence the overall number of iterations performed by the algorithm. In each iteration, a cubic subproblem

$$m_k(s) \overset{\text{def}}{=} f_k + g_k^{\mathrm{T}} s + \tfrac{1}{2} s^{\mathrm{T}} B_k s + \frac{1}{3}\sigma_k \|s\|_{M_k}^3 \tag{3.2.1}$$

must be solved which requires additional computational cost for solving more than one linear systems. This process can be computationally expensive, even when the objective function is well-behaved and the Hessian is locally positive definite.

In a recent study, Benson and Shanno [2] demonstrated that the computation of the cubic step is equivalent to solving the linear system for a certain value of the Levenberg-Marquardt perturbation parameter. This insight enables us to determine the cubic regularization parameter in a problem-specific manner for each iteration, where it is required. This selective utilization of cubic regularization permits us to benefit from some of its theoretical properties, while avoiding the associated computational overhead. The numerical results of this study offer a promising approach for improving the efficiency of cubic regularization methods while maintaining their favorable theoretical results. By utilizing the cubic regularization only when necessary, the computational cost can be reduced, thereby making the overall method more efficient. Their algorithm highlights the importance of considering the interplay between the cubic regularization and the Levenberg-Marquardt perturbation in the optimization process.

Recently, the relationship between the step calculated using a Newton-like method and the step computed using a cubic regularized model was studied by Bergou, Diouane and Gratton [3]. They introduced and evaluated a line-search framework that uses an iteration dependent ellipsoid norm in cubic regularization method. The implementation of this adaptive ellipsoid norm results in a method that behaves as a quasi-Newton algorithm with a special backtracking line search strategy. Under certain assumptions, their algorithms achieve the same convergence properties and iteration complexity as ARC algorithm.

The incorporation of the cubic regularization method with Newton's method is motivated by the desire to reduce the computational complexity associated with the minimization of the cubic subproblem. In cases where the objective function is well-behaved and the Hessian is positive definite in a local neighborhood, the Newton direction can provide robust and efficient performance without the need for regularization. Given these considerations, it is beneficial to consider using cubic regularization only in iterations where negative curvature is encountered or when there is no sufficient descent direction, rather than using it in every iteration. By carefully choosing when to employ cubic regularization, we can potentially achieve a balance between the efficiency of the Newton's method and the robustness provided by regularization.

3.2.2 The Proposed Hybrid Approaches

In this section, two different hybrid approaches of the cubic regularization method and Newton's method will be described and discussed. These hybrid approaches aim to utilize the strengths of both methods to achieve a more efficient and effective solution. Throughout this section, the exact Hessian $\nabla^2 f_k$ is used as B_k in each iteration.

Utilizing simple Newton's steps in place of the cubic step is appealing as it avoids the need to solve the cubic subproblem. As explained in Section 2.3, solving the cubic subproblem typically requires computationally expensive matrix factorizations or iterative procedures for solving a series of linear systems. When the Hessian matrix is sufficiently positive definite, i.e. the smallest eigenvalue uniformly bounded away from zero, the Newton's direction computed from the Newton's equation

$$B_k p_k = -g_k \tag{3.2.2}$$

is known to be a descent direction and is the global minimizer of a local quadratic model at x_k. The Newton's equation can be solved efficiently using various factorization methods, such as Cholesky factorization, or using iterative methods for approximate solutions. Additionally, Newton's method has fast local convergence for well-behaved functions. By replacing the cubic step with a Newton's step when possible, the number of linear systems that need to be solved can be reduced, leading to a significant reduction in computational effort.

Algorithm 3.2. Schematic outline of ARC-Newton1.

```
1: function ARC-NEWTON1
2:     Initialization: Given $x_0$, $\epsilon_d > 0$, $\gamma_2 \geq \gamma_1 > 1$, $1 > \eta_2 \geq \eta_1 > 0$, $\sigma_0 > 0$, $k = 0$.
3:     while not converged do
4:         Set cubic_step = true.
5:         if $B_k$ is positive definite then
6:             Compute an approximate solution $p_k$ of the linear system
                   $$B_k p_k = -g_k.$$
7:             if $|g_k^{\mathrm{T}} p_k^Q| \geq \epsilon_d \|g_k\| \|p_k^Q\|$ then
8:                 Search for an $\alpha_k$ satisfying the strong Wolfe conditions:
                   $$\begin{aligned} f(x_k + \alpha_k p_k) &\leq f_k + \eta_s \alpha_k g_k^{\mathrm{T}} p_k, & \eta_s &\in (0, \tfrac{1}{2}), \\ |g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k| &\leq \eta_w |g_k^{\mathrm{T}} p_k|, & \eta_w &\in (\eta_s, 1). \end{aligned}$$
9:                 if $\alpha_k > 0$ then
10:                    Update $x_{k+1} = x_k + \alpha_k p_k$.
11:                    Set cubic_step = false.
12:                end if
13:            end if
14:        end if
15:        if cubic_step is true then
16:            Compute a step $p_k$ for which
                   $$m_k(p_k) \leq m_k(p_k^C),$$
               where the Cauchy step
                   $$p_k^C = -\alpha_k^C g_k \text{ and } \alpha_k^C = \operatorname*{argmin}_{\alpha \in \mathbb{R}_+} m_k(-\alpha_k g_k).$$
17:            Compute the ratio $\rho_k = \dfrac{f_k - f(x_k + p_k)}{f_k - m_k(p_k)}$.
18:            Update
                   $$x_{k+1} = \begin{cases} x_k + p_k & \text{if } \rho \geq \eta_1, \\ x_k & \text{otherwise.} \end{cases}$$
19:        end if
20:        Update $\sigma_k$ properly.
21:        $k = k + 1$.
22:    end while
23: end function
```

The first variant of the ARC-Newton hybrid method is described in detail in

Algorithm 3.2. To determine whether the the smallest eigenvalue of Hessian is uniformly bounded away from zero, the algorithm uses a routine *chol*$(\cdot)$ to perform a Cholesky factorization of the matrix B_k. If the Cholesky factorization yields a positive definite matrix, the search direction can be calculated from Newton's equation (3.2.2) using the factorization, and then a line-search technique with strong Wolfe conditions can be applied to this Newton step. The strong Wolfe conditions ensure that the step size satisfies both the Armijo condition,

$$f(x_k + \alpha_k p_k) \leq f_k + \eta_s \alpha_k g_k^{\mathrm{T}} p_k, \quad \text{where } \eta_s \in (0, \tfrac{1}{2}), \tag{3.2.3}$$

which requires that the step satisfies a sufficient decrease condition in the objective function, and the curvature condition,

$$|g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k| \leq \eta_w |g_k^{\mathrm{T}} p_k|, \quad \text{where } \eta_w \in (\eta_s, 1), \tag{3.2.4}$$

which ensures a sufficient increase of the gradient and that the step length is not too short. Since the Hessian matrix B_k is sufficiently positive definite in this scenario, the Wolfe line search is guaranteed to give a positive step size α_k and ensures convergence. On the other hand, if the Hessian is not positive definite, it is necessary to regularize the Hessian B_k or the local quadratic model $m_k^Q(p)$. In such cases, an approximated solution of the cubic subproblem is calculated and used as the search direction.

After taking the cubic or Newton step with Wolfe line search in the ARC algorithm, the regularization parameter σ_k is updated accordingly. For the cubic step, the update is based on the agreement between the objective function and the local cubic model, as explained in Section 2.3. In the context of the Newton step combined with Wolfe line search, the approach for updating the regularization parameter is inspired by the biased Wolfe trust-region method introduced by Gertz [33]. This update relies on the search step length, denoted as α_k, and the ratio of actual to anticipated reductions, represented by ρ_k. Specifically, the ratio ρ_k is computed as

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

where p_k represents the search step taken prior to the Wolfe line search, and m_k denotes the local model. It is noteworthy that m_k can be chosen as either the local quadratic model or the local cubic model. In our numerical experiments, we did not observe significant differences between these two choices. If ρ_k exceeds a positive constant η_2 and the step length α_k is greater than or equal to a small positive constant $\alpha_{\min}$, which is between 0 and 1, the regularization parameter is updated as

$$\sigma_{k+1} = \min\left\{\frac{\sigma_k}{\alpha_k},\ \delta_1 \sigma_k\right\},$$

where δ_1 is a parameter between 0 and 1. This update ensures that the regularization parameter does not increase when a small step is taken with $\alpha_k < 1$. On the other hand, if ρ_k is less than or equal to η_2 or $\alpha_k < \alpha_{\min}$, the regularization parameter is simply updated as

$$\sigma_{k+1} = \frac{\sigma_k}{\alpha_k}$$

Even when $\rho_k \leq \eta_2$, it is possible to find a step length α_k greater than 1 through Wolfe line search, and subsequently decrease the regularization parameter. Finally, the regularization parameter is bounded by $\sigma_{\min}$ and $\sigma_{\max}$ to prevent it from becoming too small or too large.

Algorithm 3.3 outlines the second version of the ARC-Newton hybrid approach, which avoids the need to compute a cubic step even in instances where the Hessian B_k is not positive definite.

Algorithm 3.3. Schematic outline of ARC-Newton2.

```
1: function ARC-NEWTON2
2:     Initialization: Given $x_0$, $\epsilon_d > 0$, $\gamma_2 \geq \gamma_1 > 1$, $1 > \eta_2 \geq \eta_1 > 0$, $\sigma_0 > 0$, $k = 0$.
3:     while not converged do
4:         Set cubic_step = true
5:         Compute an approximate solution $p_k^Q$ of the linear system
               $$B_k p_k = -g_k.$$
6:         if $B_k$ is not ill-conditioned and $|g_k^{\mathrm{T}} p_k^Q| \geq \epsilon_d \|g_k\| \|p_k^Q\|$ then
7:             Set
               $$p_k = -\operatorname{sign}(g_k^{\mathrm{T}} p_k^Q) p_k^Q.$$
8:             Search for an $\alpha_k$ satisfying the strong Wolfe conditions:
               $$\begin{aligned} f(x_k + \alpha_k p_k) &\leq f_k + \eta_s \alpha_k g_k^{\mathrm{T}} p_k, & \eta_s \in (0, \tfrac{1}{2}), \\ |g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k| &\leq \eta_w |g_k^{\mathrm{T}} p_k|, & \eta_w \in (\eta_s, 1). \end{aligned} \tag{3.2.5}$$
9:             if $\alpha_k > 0$ then
10:                Update $x_{k+1} = x_k + \alpha_k p_k$.
11:                Set cubic_step = false.
12:            end if
13:        end if
14:        if cubic_step is true then
15:            Compute a step $p_k$ for which
               $$m_k(p_k) \leq m_k(p_k^C),$$
               where the Cauchy step
               $$p_k^C = -\alpha_k^C g_k \text{ and } \alpha_k^C = \operatorname*{argmin}_{\alpha \in \mathbb{R}_+} m_k(-\alpha_k g_k).$$
16:            Compute the ratio $\rho_k = \dfrac{f_k - f(x_k + p_k)}{f_k - m_k(p_k)}$.
17:            Update
               $$x_{k+1} = \begin{cases} x_k + p_k & \text{if } \rho \geq \eta_1, \\ x_k & \text{otherwise.} \end{cases}$$
18:        end if
19:        Update $\sigma_k$ properly.
20:        $k = k + 1$.
21:    end while
22: end function
```

In contrast to Algorithm 3.2, in Algorithm 3.3, a trial step p_k^Q is always computed using the Newton's equation (3.2.2), irrespective of the behavior of the local Hessian matrix B_k. If the absolute value of the directional derivative of trial step p_k^Q and the gradient g_k is bounded away from zero, the search direction is determined by either setting it to the trial step p_k^Q or the negated trial step $-p_k^Q$, depending on the sign of the directional derivative. To ensure convergence, a line search method with strong Wolfe conditions, as specified by (3.2.5), is applied on the search direction to acquire a step size α_k. Conversely, if a sufficient descent direction cannot be obtained from the solution p_k^Q of Newton's equation, the search direction is instead set to an approximated solution of the cubic subproblem. The acceptance of the search step hinges on the agreement between the objective function and the local cubic model, as defined in ρ_k. Upon finding a satisfactory agreement, the search step is accepted. The regularization parameter σ_k is subsequently adjusted accordingly, as discussed in Section 2.3 and Algorithm 3.2.

3.2.3 Theoretical Discussion

This section examines the relationship between the Newton direction and the cubic step, and presents a convergence analysis of the ARC-Newton hybrid algorithms.

The subsequent finding demonstrates that, provided the quasi-Newton direction is not orthogonal to the gradient of the objective function at the current iteration, the Newton direction is collinear with the solution of a certain cubic model determined by a specific selection of the ellipsoid norm.

Theorem 3.2.1. *Suppose the exact solution p_k^Q of Newton's equation* (3.2.2) *is not orthogonal to the gradient of the objective function at x_k, i.e.,*

$$|g_k^{\mathrm{T}} p_k^Q| \geq \epsilon_d \|g_k\| \|p_k^Q\|, \tag{3.2.6}$$

with ϵ_d being a predetermined positive constant. Then, a positive scalar δ_k exists such that

$$p_k^Q = \delta_k p_k^C, \tag{3.2.7}$$

where p_k^C represents a global minimizer of a certain cubic model defined by (3.2.1).

Proof. Given that p_k^Q fulfills (3.2.6), according to Theorem 4.1 in [3], a symmetric positive definite matrix M_k exists such that

$$M_k p_k^Q = \frac{\|p_k^Q\|M_k}{g_k^{\mathrm{T}} p_k^Q} g_k.$$

By employing the ellipsoid norm $\|\cdot\|_{M_k}$ in the cubic model m_k definition, as defined in (3.2.1), in conjunction with Theorem 3.3 from [3], we can deduce that an approximate optimal point of subproblem (3.2.1) takes the form

$$p_k^C = \frac{1}{\delta_k} p_k^Q, \quad \text{where} \quad \delta_k = \frac{1}{2}\left(1 - \operatorname{sign}(g_k^{\mathrm{T}} p_k^Q)\sqrt{1 + 4\frac{\sigma_k \|p_k^Q\|_{M_k}^3}{|g_k^{\mathrm{T}} p_k^Q|}}\right).$$

□

Next, we examine the step taken on the Wolfe line search. The following lemma establishes that any step that satisfies the strong Wolfe conditions is a step of sufficient decrease.

Lemma 3.2.1. *Let f be a scalar-valued, twice-continuously differentiable function defined on an open convex set $\mathcal{D} \subset \mathbb{R}^n$. Suppose $x_0 \in \mathcal{D}$ is selected such that the level set $\mathcal{L}(f_0)$ is closed and bounded. Assume that p_k is a search direction satisfying $g(x_k)^{\mathrm{T}} p_k < 0$ and $\|p_k\| \le \gamma$, where γ is a constant independent of k. Then, the search step p_k that meets the strong Wolfe conditions,* (3.2.3) *and* (3.2.4)*, is a step length of sufficient decrease, i.e.,*

$$\lim_{k\to\infty} |g_k^{\mathrm{T}} p_k| = 0. \tag{3.2.8}$$

Proof. To prove this theorem, we employ a proof by contradiction. Assume there exists a constant $\epsilon > 0$ such that $|g_k^{\mathrm{T}} p_k| \geq \epsilon$ for infinitely often. Note that the backtracking condition (3.2.3) infers that

$$f_k - f_{k+1} \geq \eta_s \alpha_k |g_k^{\mathrm{T}} p_k|.$$

This relationship suggests that x_k is well-defined and persists within $\mathcal{L}(f_0)$. The assumption of f being bounded below on $\mathcal{L}(f_0)$ leads to the conclusion that f_k represents a bounded, strictly decreasing sequence, which consequently converges.

Let $\mathcal{K} = \{k : |g_k^{\mathrm{T}} p_k| \geq \epsilon\}$. The first Wolfe condition (3.2.3) yields

$$f_k - f_{k+1} \geq \eta_s \alpha_k \epsilon, \quad \text{for all } k \in \mathcal{K}. \tag{3.2.9}$$

Given that $\{f_k\}$ is a converging sequence, the left-hand side of (3.2.9) approaches zero for $k \in \mathcal{K}$. The definitions of η_s and ϵ imply that $\alpha_k \to 0$ for $k \in \mathcal{K}$, and the uniform boundedness of the sequence $\{p_k\}$ establishes $\alpha_k p_k \to 0$ for $k \in \mathcal{K}$.

By rearranging the second Wolfe condition (3.2.4) and given that $|g_k^{\mathrm{T}} p_k| = -g_k^{\mathrm{T}} p_k$, we can deduce that

$$g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k - g_k^{\mathrm{T}} p_k \geq (1 - \eta_w)|g_k^{\mathrm{T}} p_k| \geq (1 - \eta_w)\epsilon, \quad \text{for all } k \in \mathcal{K}. \tag{3.2.10}$$

Utilizing standard norm inequalities, we derive

$$g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k - g_k^{\mathrm{T}} p_k \leq \|g(x_k + \alpha_k p_k) - g_k\|_D \|p_k\|. \tag{3.2.11}$$

where $\|\cdot\|_D$ represents the norm dual to $\|\cdot\|$. Inequalities (3.2.10) and (3.2.11) imply that

$$\|g(x_k + \alpha_k p_k) - g_k\|_D \|p_k\| \geq (1 - \eta_w)\epsilon > 0, \quad \text{for all } k \in \mathcal{K}. \tag{3.2.12}$$

Since $1 - \eta_w$ and ϵ are bounded away from zero, this inequality indicates that the vector difference within the norm on the left side is bounded away from zero. However, we recognize that $\alpha_k p_k \to 0$ for all $k \in \mathcal{K}$, and the continuity of $g(x)$ along with the boundedness of p_k imply that the left-hand side of (3.2.12) converges to zero. This leads to the intended contradiction, demonstrating that $|g_k^{\mathrm{T}} p_k| \to 0$. □

Now, we are ready to prove the first-order convergence result of Algorithm 3.2. The convergence results of Algorithm 3.3 can be derived analogously.

Theorem 3.2.2. *Let $\{x_k\}$ be the sequence generated by Algorithm* 3.2*. Suppose $x_0 \in \mathcal{D}$ is selected such that the level set $\mathcal{L}(f_0)$ is closed and bounded. If $\{f_k\}$ is bounded below and f and g are uniformly continuous on the sequence $\{x_k\}$, then either there is a finite k such that $g_k = 0$, or*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.2.13}$$

Proof. Let us begin by assuming that only finitely many Newton steps are taken. In this case, Theorem 2.3.1 implies that the result (3.2.13) holds. Now, suppose that infinitely many Newton steps are taken. If $g_k = 0$ for some finite index k, then the theorem holds trivially. Therefore, we may assume that $g_k \neq 0$ for all k.

If $x_k \in \mathcal{L}(f_0)$, the Hessian B_k being positive-definite ensures that the Newton direction is always a direction of descent. Lemma 3.1.3 then guarantees the existence of a suitable α_k satisfying the Wolfe conditions (3.2.3) and (3.2.4), which implies that $x_{k+1} \in \mathcal{L}(f_0)$. This shows that the iterates are well-defined.

From Lemma 3.2.1, we have demonstrated that the Newton's step generated by Algorithm 3.2 satisfying the strong Wolfe conditions has the property that

$$\lim_{k\to 0} |g_k^{\mathrm{T}} p_k| = 0. \tag{3.2.14}$$

To achieve convergence, it is necessary to guarantee that $|g_k^{\mathrm{T}} p_k|$ approaches zero only when

g_k tends to zero.

Because of the Newton's equation (3.2.2), we can obtain that

$$|g_k^{\mathrm{T}} p_k| = |p_k^{\mathrm{T}} B_k p_k| \geq \lambda_{\min} \|p_k\|^2. \tag{3.2.15}$$

Applying standard norm inequalities to (3.2.2) yields

$$\|p_k\| \geq \frac{\|g_k\|}{\|B_k\|}. \tag{3.2.16}$$

Using the relation $\|B_k\| = \lambda_{\max}$, we can combine (3.2.15) and (3.2.16) as follows:

$$|g_k^{\mathrm{T}} p_k| \geq \frac{\lambda_{\min}}{\lambda_{\max}} \|p_k\| \|g_k\| \geq \frac{\lambda_{\min}}{\lambda_{\max}^2} \|g_k\|^2. \tag{3.2.17}$$

By the construction of the algorithm, the smallest eigenvalue $\lambda_{\min}$ of B_k has a lower bound. Meanwhile, the continuity of the Hessian and the compactness of $\mathcal{L}(f_0)$ together imply that $\|B_k\|$ is bounded, which means the largest eigenvalue $\lambda_{\max}$ has an upper bound. Therefore, using (3.2.17), together with (3.2.14), we obtain $g_k \to 0$. $\square$

Chapter 4

Cubic Regularization Algorithms for Constrained Optimization

4.1 Introduction

In this chapter, we consider a primal-dual adaptive cubic regularization algorithm (PDARC) for solving the inequality-constrained nonlinear programming problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad c(x) \geq 0, \tag{NIP}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $f : \mathbb{R}^n \mapsto \mathbb{R}$ are twice-continuously differentiable nonlinear functions. Interior methods are a family of algorithms that solve (NIP) by transforming the constrained problem into a parameterized unconstrained problem using penalty and barrier functions.

The proposed method is based on minimizing a shifted primal-dual penalty-barrier merit function introduced by Gill, Kungurtsev and Robinson [22]. This method has a two-level structure of inner and outer iterations. The inner iterations are those of a method for unconstrained minimization used to find an approximate minimizer of a merit function that represents a compromise between minimizing the objective function and satisfying the constraints. The merit function is the sum of three terms: the objective function, a penalty term for the equality constraints, and a barrier term for the inequality constraints. For a given unconstrained minimization, the merit function is defined in terms of fixed values of a penalty parameter, a barrier parameter and estimates of the Lagrange multipliers. The outer iterations measure and ensure progress toward a solution by adjusting the penalty-barrier parameters and Lagrange multiplier estimates.

In this chapter, we propose an efficient convergent cubic regularization method for generating the inner iterates. A cubic regularization technique combined with a line-search method is applied to handle local nonconvexity and ensure global convergence. Moreover, we show that the solution of the cubic regularized subproblem can be obtained by solving an equivalent system that has the same structure as the conventional primal-dual system, which implies that the corresponding linear systems can be solved by efficient off-the-shelf linear system-solvers. Hence the proposed method can be scaled directly to large-scale problems.

Section 4.2 is devoted to a brief overview and the motivation for the application of primal-dual cubic regularization to constrained nonlinear optimization. In Section 4.3, we present a cubic regularization method that finds an approximate minimizer of the shifted penalty-barrier merit function in the inner iteration, and show that the cubic regularized subproblem can be solved by factoring matrices with the same structure as conventional primal-dual matrices. In Section 4.4, we describe an adaptive cubic regularization method that uses a practical Armijo-type line search and its variants. The method maintains feasibility of primal and dual variables, and utilizes the trial step computed from the cubic subproblem to improve a primal-dual merit function at each iteration. In Section 4.5, the convergence results of the proposed algorithm are established under certain assumptions. In Section 4.6, a practical algorithm based on a modified version of the Moré-Sorensen method [34] is proposed for the solution of the primal-dual cubic subproblem. Section 4.7 concerns the application of the proposed method for solving constrained optimization problems in the general form, and additional information regarding the derivation of the relevant equations and solutions is available in Appendix A.

4.2 Algorithm Overview and Background

The problem (NIP) can be reformulated by introducing non-negative slack variables for the nonlinear inequality constraints. This gives the equivalent problem

$$\underset{x\in\mathbb{R}^n, s\in\mathbb{R}^m}{\text{minimize}}\ f(x) \quad \text{subject to} \quad c(x)-s=0, \quad s\geq 0. \tag{4.2.1}$$

The vector (x^*, s^*, y^*, w^*) is called a first-order KKT point for problem (4.2.1) if the following KKT conditions hold

$$\left.\begin{aligned} \nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* &= 0, & & \\ y^* - w^* &= 0, & w^* &\geq 0, \\ c(x^*) - s^* &= 0, & s^* &\geq 0, \\ w^* \cdot s^* &= 0. & & \end{aligned}\right\} \tag{4.2.2}$$

Under certain regularity conditions, the KKT conditions are first-order conditions for an optimal solution of problem (4.2.1). The vectors y^* and w^* constitute the Lagrange multiplier vectors for, respectively, the equality constraints $c(x) - s = 0$ and the inequality constraints $s \geq 0$. Let the vector (x, s, y, w) denote the k-th primal-dual iterate. The outer iteration of PDARC aims to ensure that the sequence $\{x_k, s_k, y_k, w_k\}$ approximately follows a continuous primal-dual trajectory and converges to a point that satisfies the KKT conditions (4.2.2). To design an efficient algorithm, the conditions (4.2.2) are perturbed as follows:

$$\left.\begin{aligned} \nabla f(x) - J(x)^{\mathrm{T}} y &= 0, & & \\ y - w &= 0, & w &\geq 0, \\ c(x) - s &= \mu^P(y^E - y), & s &\geq 0, \\ w \cdot s &= \mu^B(w^E - w), & & \end{aligned}\right\} \tag{4.2.3}$$

where $y^E \in \mathbb{R}^m$ is an estimate of a Lagrange multiplier vector for the constraints $c(x)-s=0$, $w^E \in \mathbb{R}^m$ is an estimate of a Lagrange multiplier for the constraints $s \geq 0$, and the scalars μ^P and μ^B are positive penalty and barrier parameters, respectively. The perturbed conditions (4.2.3) motivate the following merit function for a path-following interior method. The shifted primal-dual problem associated with problem (4.2.1) is obtained by including the constraints $c(x) - s = 0$ with the objective using a shifted primal-dual augmented Lagrangian term. A shifted primal-dual penalty-barrier term is used for the simple bounds. This gives the problem

$$\underset{x,s,y,w}{\text{minimize}} \ \mathcal{M}(x,s,y,w\,;\mu^P,\mu^B,y^E,w^E) \quad \text{subject to} \ \ s+\mu^B e > 0, \quad w > 0, \tag{4.2.4}$$

where $\mathcal{M}(x,s,y,w\,;\mu^P,\mu^B,y^E,w^E)$ is the shifted primal-dual penalty-barrier function

$$\begin{aligned}\mathcal{M}(x,s,y,w\,;\mu^P,\mu^B,y^E,w^E) &= f(x) - \big(c(x)-s\big)^{\mathrm{T}} y^E \\ &\quad + \frac{1}{2\mu^P}\|c(x)-s\|^2 + \frac{1}{2\mu^P}\|c(x)-s+\mu^P(y-y^E)\|^2 \\ &\quad - \sum_{i=1}^{m} \mu^B w_i^E \ln\big(s_i+\mu^B\big) - \sum_{i=1}^{m} \mu^B w_i^E \ln\big(w_i(s_i+\mu^B)\big) + \sum_{i=1}^{m} w_i(s_i+\mu^B),\end{aligned}$$

which is well defined for all w and s such that $w > 0$ and $s + \mu^B e > 0$. It can be shown that the Newton equations for finding a zero of the perturbed conditions (4.2.3) are equivalent to certain approximate Newton equations for finding a minimizer of $\mathcal{M}$ in the neighborhood of a second-order minimizer of the problem (4.2.1). Under certain assumptions, the primal-dual iteration attempts to follow a differentiable trajectory that converges to a limit point that is either an infeasible stationary point or a complementary approximate KKT point. For further details see Gill, Kungurtsev and Robinson [22].

To handle nonconvexity, Gill, Kungurtsev and Robinson [22] apply an inertia-controlling symmetric indefinite factorization (for more details, see Forsgren [17]). However, this inertia-controlling factorization interchanges certain rows and columns, which interferes with the row and column ordering used to maintain sparsity in the factors. This hinders the application of state-of-the-art software, or software developed for specific types of advanced architectures. To overcome this issue, the cubic regularization method is used to minimize $\mathcal{M}(x, s, y, w\,; \mu^{P}, \mu^{B}, y^{E}, w^{E})$, and is discussed in the following sections.

4.3 Primal-Dual Cubic Regularization Methods

In this section, we focus on an inner iteration and discuss how the cubic regularization method can be used to find an approximate minimizer of the shifted penalty-barrier merit function $\mathcal{M}(x, s, y, w; \mu^{P}, \mu^{B}, y^{E}, w^{E})$ for fixed $(\mu^{P}, \mu^{B}, y^{E}, w^{E})$. Let c, g and J denote the quantities $c(x)$, $\nabla f(x)$ and $J(x)$. For clarity, the dependence of $\mathcal{M}$ on the parameters μ^{P}, μ^{B}, y^{E} and w^{E}, will be suppressed when appropriate, with $\mathcal{M}$ or $\mathcal{M}(x, s, y, w)$ being used to denote $\mathcal{M}(x, s, y, w\,; \mu^{P}, \mu^{B}, y^{E}, w^{E})$. Let S^{μ} and W denote diagonal matrices with diagonal entries $s_i + \mu^{B}$ and w (i.e., $S^{\mu} = \operatorname{diag}(s + \mu^{B})$ and $W = \operatorname{diag}(w)$) such that $s_i + \mu^{B} > 0$ and $w_i > 0$. It is convenient to define the positive-definite matrices

$$D_{Y} = \mu^{P} I_m \qquad \text{and} \qquad D_{W} = S^{\mu} W^{-1},$$

and auxiliary vectors

$$\pi^{Y} = \pi^{Y}(x, s) = y^{E} - \frac{1}{\mu^{P}}(c - s) \quad \text{and} \quad \pi^{W} = \pi^{W}(s) = \mu^{B}(S^{\mu})^{-1} w^{E}.$$

In order to facilitate global convergence, we apply a cubic regularization method based on finding an approximate solution of the primal-dual cubic subproblem

$$\underset{\Delta v \in \mathbb{R}^{n+3m}}{\text{minimize}} \;\; \mathcal{C}(\Delta v) = \mathcal{M}(v) + \nabla\mathcal{M}(v)^{\mathrm{T}} \Delta v + \frac{1}{2} \Delta v^{\mathrm{T}} B(v) \Delta v + \frac{1}{3} \mu^{C} \|\Delta v\|_{T}^{3}, \tag{4.3.1}$$

where $v = (x,\, s,\, y,\, w)$, $\Delta v = (\Delta x,\, \Delta s,\, \Delta y,\, \Delta w)$, $\nabla\mathcal{M}$ is the gradient of the merit function, and B is an approximation of the Hessian of the merit function. The cubic term involves the elliptic norm $\|\Delta v\|_T = (\Delta v^{\mathrm{T}} T \Delta v)^{1/2}$, and the nonnegative regularization parameter $\mu^C \geq 0$. The matrix T is a positive-definite diagonal of the form $T = \operatorname{diag}(T^x, T^s, T^y, T^w)$. The matrix $B = B(v)$ and vector $\nabla\mathcal{M} = \nabla\mathcal{M}(v)$ are given by

$$B = \begin{pmatrix} H + 2J^{\mathrm{T}} D_Y^{-1} J & -2J^{\mathrm{T}} D_Y^{-1} & J^{\mathrm{T}} & 0 \\ -2D_Y^{-1} J & 2\big(D_Y^{-1} + D_W^{-1}\big) & -I_m & I_m \\ J & -I_m & D_Y & 0 \\ 0 & I_m & 0 & D_W \end{pmatrix}$$

and

$$\nabla\mathcal{M} = \begin{pmatrix} g - J^{\mathrm{T}}\big(\pi^Y + (\pi^Y - y)\big) \\ (2\pi^Y - y) - (2\pi^W - w) \\ -D_Y(\pi^Y - y) \\ -D_W(\pi^W - w) \end{pmatrix},$$

where $H = H(x,y)$, $g = \nabla f(x)$, and $J = J(x)$.

The following theorem provides the theoretical basis for finding an approximate global solution of the cubic regularized subproblem under some predefined positive-definite diagonal matrix T.

Theorem 4.3.1 (Optimality Conditions for the Cubic Model). *Any Δv^* is a global minimizer of the cubic subproblem* (4.3.1) *over $\mathbb{R}^{n+3m}$ if and only if there exists a unique $\sigma \geq 0$ such that*

$$(B + \sigma T)\Delta v^* = -\nabla\mathcal{M}, \quad \frac{\sigma}{\mu^C} = \|\Delta v^*\|_T,$$

with $B + \sigma T$ positive semidefinite. For any global minimizer Δv^*, the value of σ is unique and independent of Δv^*. Moreover, if $B + \sigma I$ is positive definite, Δv^* is unique.

The application of the method of Moré and Sorensen [34] to solve the subproblem

(4.3.1) requires seeking an optimal σ by calculating the Cholesky factorization of $B + \sigma T$ for various values of σ. Instead of factorizing $B + \sigma T$ directly, with appropriate choices of T, the solution of $(B + \sigma T)\Delta v = -\nabla \mathcal{M}$ can be obtained by solving an equivalent system that has the same primal-dual structure. The equivalence can be established as follows. Let's write the gradient and Hessian of the merit function in terms of the vector $\bar{x} = (x, s)$ and $\bar{y} = (w, y)$. Let $\bar{g}$, $\bar{H}$, $\bar{J}$ and $\bar{D}$ denote the quantities

$$\bar{g} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \quad \bar{H} = \begin{pmatrix} H & 0 \\ 0 & 0 \end{pmatrix}, \quad \bar{J} = \begin{pmatrix} J & -I_m \\ 0 & I_m \end{pmatrix}, \text{ and } \bar{D} = \begin{pmatrix} D_Y & 0 \\ 0 & D_W \end{pmatrix}.$$

Similarly, let $\bar{T}_x = \operatorname{diag}(T_x, T_s)$ and $\bar{T}_y = \operatorname{diag}(T_y, T_w)$. The equations $(B + \sigma T)p = -\nabla \mathcal{M}$ may be written in the form

$$\begin{pmatrix} \bar{H} + 2\bar{J}^{\mathrm{T}}\bar{D}^{-1}\bar{J} + \sigma \bar{T}_x & \bar{J}^{\mathrm{T}} \\ \bar{J} & \bar{D} + \sigma \bar{T}_y \end{pmatrix} \begin{pmatrix} \Delta \bar{x} \\ \Delta \bar{y} \end{pmatrix} = -\begin{pmatrix} \bar{g} - \bar{J}^{\mathrm{T}}\bar{\pi} - \bar{J}^{\mathrm{T}}(\bar{\pi} - \bar{y}) \\ -\bar{D}(\bar{\pi} - \bar{y}) \end{pmatrix}, \qquad (4.3.2)$$

where

$$\bar{\pi} = \begin{pmatrix} \pi^{Y} \\ \pi^{W} \end{pmatrix}, \quad \Delta \bar{x} = \begin{pmatrix} \Delta x \\ \Delta s \end{pmatrix}, \text{ and } \Delta \bar{y} = \begin{pmatrix} \Delta y \\ \Delta w \end{pmatrix}.$$

Applying the nonsingular matrix

$$\begin{pmatrix} I & -2\bar{J}^{\mathrm{T}}\bar{D}^{-1} \\ 0 & I \end{pmatrix}$$

to both sides of equation (4.3.2) gives the system

$$\begin{pmatrix} \bar{H} + \sigma \bar{T}_x & -\bar{J}^{\mathrm{T}}(I + 2\sigma \bar{D}^{-1}\bar{T}_y) \\ \bar{J} & \bar{D} + \sigma \bar{T}_y \end{pmatrix} \begin{pmatrix} \Delta \bar{x} \\ \Delta \bar{y} \end{pmatrix} = -\begin{pmatrix} \bar{g} - \bar{J}^{\mathrm{T}}\bar{y} \\ \bar{D}(\bar{y} - \bar{\pi}) \end{pmatrix}.$$

With the substitutions $\bar{T}_x = I$ and $\bar{T}_y = \bar{D}$ and some simplification, the quantities $\Delta\bar{x}$ and $\Delta\bar{y}$ may be obtained by solving the equations

$$\begin{pmatrix} \bar{H} + \sigma I & -\bar{J}^{\mathrm{T}} \\ \bar{J} & \bar{\sigma}\bar{D} \end{pmatrix} \begin{pmatrix} \Delta\bar{x} \\ (1+2\sigma)\Delta\bar{y} \end{pmatrix} = -\begin{pmatrix} \bar{g} - \bar{J}^{\mathrm{T}}\bar{y} \\ \bar{D}(\bar{y} - \bar{\pi}) \end{pmatrix}, \tag{4.3.3}$$

where $\bar{\sigma} = (1+\sigma)/(1+2\sigma)$. In terms of the original variables, the unsymmetric equations (4.3.3) may be written as

$$\begin{pmatrix} H + \sigma I_n & 0 & -J^{\mathrm{T}} & 0 \\ 0 & \sigma I_m & I_m & -I_m \\ J & -I_m & \bar{\sigma} D_Y & 0 \\ 0 & I_m & 0 & \bar{\sigma} D_W \end{pmatrix} \begin{pmatrix} \mathit{\Delta} x \\ \mathit{\Delta} s \\ (1+2\sigma)\mathit{\Delta} y \\ (1+2\sigma)\mathit{\Delta} w \end{pmatrix} = -\begin{pmatrix} g - J^{\mathrm{T}} y \\ y - w \\ D_Y(y - \pi^Y) \\ D_W(w - \pi^W) \end{pmatrix}. \tag{4.3.4}$$

With our choice of T, the matrix in (4.3.4) has the same sparsity pattern as the matrix in primal-dual equations. Equation (4.3.4) can be symmetrized by collecting the factor $-(1+2\sigma)$ into Δy and Δw, which gives the equivalent symmetric system

$$\begin{pmatrix} H + \sigma I_n & 0 & J^{\mathrm{T}} & 0 \\ 0 & \sigma I_m & -I_m & I_m \\ J & -I_m & \bar{\sigma} D_Y & 0 \\ 0 & I_m & 0 & \bar{\sigma} D_W \end{pmatrix} \begin{pmatrix} \mathit{\Delta} x \\ \mathit{\Delta} s \\ \mathit{\Delta}\widetilde{y} \\ \mathit{\Delta}\widetilde{w} \end{pmatrix} = -\begin{pmatrix} g - J^{\mathrm{T}} y \\ y - w \\ D_Y(y - \pi^Y) \\ D_W(w - \pi^W) \end{pmatrix}, \tag{4.3.5}$$

with $\mathit{\Delta}\widetilde{y} = -(1+2\sigma)\mathit{\Delta} y$ and $\mathit{\Delta}\widetilde{w} = -(1+2\sigma)\mathit{\Delta} w$. The linear equation (4.3.5) is a symmetric and well-scaled system in which $\bar{\sigma} D_Y$ and $\bar{\sigma} D_W$ have the role of regularizing the KKT matrix. After the application of block elimination and the collection of terms, the solution of the equations (4.3.5) can be obtained from an equivalent linear system of smaller size. Let $\breve{D}_W$ denote the $m \times m$ diagonal matrix

$$\breve{D}_W = (D_W^{-1} + \sigma\bar{\sigma} I_m)^{-1}.$$

The solution of (4.3.5) can be written as

$$\begin{aligned}\mathit{\Delta} s &= -\bar{\sigma}\breve{D}_W\Big(y - \mathit{\Delta}\widetilde{y} - w + \frac{1}{\bar{\sigma}}(w - \pi^W)\Big),\\ \mathit{\Delta}\widetilde{w} &= -D_W^{-1}\breve{D}_W\Big(y - \mathit{\Delta}\widetilde{y} - w - \sigma D_W(w - \pi^W)\Big),\end{aligned}$$

where $\mathit{\Delta} x$ and $\mathit{\Delta}\widetilde{y}$ satisfy the KKT equations

$$\begin{aligned}&\begin{pmatrix} H + \sigma I_n & J^{\mathrm{T}} \\ J & -\bar{\sigma}(D_Y + \breve{D}_W) \end{pmatrix}\begin{pmatrix} \mathit{\Delta} x \\ \mathit{\Delta}\widetilde{y} \end{pmatrix}\\ &\qquad = -\begin{pmatrix} g - J^{\mathrm{T}} y \\ D_Y(y - \pi^Y) + \breve{D}_W\big(w - \pi^W + \bar{\sigma}(y - w)\big) \end{pmatrix}.\end{aligned} \tag{4.3.6}$$

The vectors $\mathit{\Delta} y$ and $\mathit{\Delta} w$ are computed from the equations

$$\mathit{\Delta} y = -\frac{1}{(1 + 2\sigma)}\mathit{\Delta}\widetilde{y} \quad \text{and} \quad \mathit{\Delta} w = -\frac{1}{(1 + 2\sigma)}\mathit{\Delta}\widetilde{w}.$$

Note that the symmetric system (4.3.6) is of order $n + m$ and does not involve the slack variables. The major cost of an iteration for finding an approximate solution of the cubic subproblem (4.3.1) is the cost of solving this system. The only extra computational costs associated with this reformulation are from updating the slack variables.

The generic equations. For generic equations of the form $(B + \sigma T)p = b$, the equations are

$$\begin{pmatrix} H+\sigma I_n & 0 & J^{\mathrm{T}} & 0 \\ 0 & \sigma I_m & -I_m & I_m \\ J & -I_m & \bar{\sigma} D_{\scriptscriptstyle Y} & 0 \\ 0 & I_m & 0 & \bar{\sigma} D_{\scriptscriptstyle W} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}. \tag{4.3.7}$$

Using block elimination, the solution of these equations is given by

$$p_4 = (I + \sigma\bar{\sigma} D_{\scriptscriptstyle W})^{-1}(p_3 - b_2 + \sigma b_4) \quad \text{and} \quad p_2 = b_4 - \bar{\sigma} D_{\scriptscriptstyle W} p_4,$$

where the vectors p_1 and p_4 satisfy the equations

$$\begin{pmatrix} H+\sigma I_n & J^{\mathrm{T}} \\ J & -\bar{\sigma}(D_{\scriptscriptstyle Y} + \breve{D}_{\scriptscriptstyle W}) \end{pmatrix} \begin{pmatrix} p_1 \\ p_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ \sigma \breve{D}_{\scriptscriptstyle W}(b_2 - \sigma b_4) + b_3 + b_4 \end{pmatrix}.$$

In order to satisfy the optimality conditions of Theorem 4.3.1, we are interested in values of σ such that $B + \sigma T$ is positive semidefinite. Instead of checking $B + \sigma T$ directly, the following lemma indicates that the inertia of $B + \sigma T$ can be deduced from the inertia of $K(\sigma)$. For simplicity, let the matrix $B + \sigma T$ be denoted by $B(\sigma)$, i.e.,

$$B(\sigma) = \begin{pmatrix} H + 2J^{\mathrm{T}} D_{\scriptscriptstyle Y}^{-1} J + \sigma I_n & -2J^{\mathrm{T}} D_{\scriptscriptstyle Y}^{-1} & J^{\mathrm{T}} & 0 \\ -2D_{\scriptscriptstyle Y}^{-1} J & 2\big(D_{\scriptscriptstyle Y}^{-1} + D_{\scriptscriptstyle W}^{-1}\big) + \sigma I_m & -I_m & I_m \\ J & -I_m & (1+\sigma) D_{\scriptscriptstyle Y} & 0 \\ 0 & I_m & 0 & (1+\sigma) D_{\scriptscriptstyle W} \end{pmatrix},$$

and let the condensed matrix in equation (4.3.6) be denoted by $K(\sigma)$, i.e.,

$$K(\sigma) = \begin{pmatrix} H+\sigma I_n & J^{\mathrm{T}} \\ J & -\bar{\sigma}(D_{\scriptscriptstyle Y} + \breve{D}_{\scriptscriptstyle W}) \end{pmatrix}.$$

Lemma 4.3.1. *The inertia of $B(\sigma)$ and $K(\sigma)$ satisfies the identities:*

$$\begin{aligned}\operatorname{In}(B(\sigma)) &= \operatorname{In}(H + \sigma I + \frac{1}{\sigma} J^{\mathrm{T}} (D_Y + \breve{D}_W)^{-1} J) + (3m, 0, 0),\\ \operatorname{In}(K(\sigma)) &= \operatorname{In}(H + \sigma I + \frac{1}{\sigma} J^{\mathrm{T}} (D_Y + \breve{D}_W)^{-1} J) + (0, m, 0).\end{aligned}$$

Proof. We will follow the result from Lemma 4.1 in Forsgren and Gill [18] to prove these relations. Consider a symmetric block-partitioned matrix X of the form

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

The inertia of X satisfies $\operatorname{In}(X) = \operatorname{In}(C) + \operatorname{In}(A - B^{\mathrm{T}} C^{-1} B)$. Both $B(\sigma)$ and $K(\sigma)$ are symmetric block-partitioned matrices. As D_Y and D_W are diagonal positive definite matrices, the inertia property implies that

$$\begin{aligned}\operatorname{In}(B(\sigma)) &= \operatorname{In}\begin{pmatrix} H + \sigma I_n + \frac{1}{\sigma} J^{\mathrm{T}} D_Y^{-1} J & -\frac{1}{\sigma} J^{\mathrm{T}} D_Y^{-1} \\ -\frac{1}{\sigma} D_Y^{-1} J & \sigma I_n + \frac{1}{\sigma}(D_Y + D_W) \end{pmatrix} + (2m, 0, 0)\\ &= \operatorname{In}(H + \sigma I + \frac{1}{\sigma} J^{\mathrm{T}} (D_Y + \breve{D}_W)^{-1} J) + (m, 0, 0) + (2m, 0, 0)\\ &= \operatorname{In}(H + \sigma I + \frac{1}{\sigma} J^{\mathrm{T}} (D_Y + \breve{D}_W)^{-1} J) + (3m, 0, 0).\end{aligned}$$

Similarly,

$$\operatorname{In}(K(\sigma)) = \operatorname{In}(H + \sigma I + \frac{1}{\sigma} J^{\mathrm{T}} (D_Y + \breve{D}_W)^{-1} J) + (m, 0, 0).$$

□

A solution of the cubic subproblem (4.3.1) can be obtained by repeatedly factorizing the matrix $K(\sigma)$ and solving the system (4.3.6) for different values of σ. However, we are particularly interested in values of σ for which $B(\sigma)$ is positive semidefinite, as stated in Theorem 4.3.1. The relationship between the inertia of $B(\sigma)$ and the inertia of $K(\sigma)$ in

Lemma 4.3.1 may be used to find values σ for which $B(\sigma)$ is positive definite. In particular, we can count the number of negative eigenvalues of $K(\sigma)$ using the symmetric indefinite factorization $P^{\mathrm{T}}K(\sigma)P = LDL^{\mathrm{T}}$, where the number of negative eigenvalues of $K(\sigma)$ is the number of 2×2 blocks and negative 1×1 blocks of D (see Bunch and Parlett [5]). If the number of negative eigenvalues of $K(\sigma)$ is larger than m, then σ must be increased and $K(\sigma)$ is refactorized. This procedure is repeated until $B(\sigma)$ is positive semidefinite.

In the case of primal-dual interior-point methods, the function being minimized includes a logarithmic barrier term that is undefined outside of the feasible region. We may expect that the cubic regularized subproblem (4.3.1) may need to be solved repeatedly in order to obtain a feasible step.

4.4 Statement of the Algorithm

This section provides the formal descriptions of the proposed primal-dual cubic regularization algorithm. Cubic regularization methods have both favorable theoretical properties and excellent numerical results in practice. However, a significant cost of cubic regularization is the need to find an approximate minimizer of the cubic regularized subproblem, which may require the solution of the system (4.3.6) several times with different values of σ. Performing a line search along the search direction takes advantage of the computational effort involved in minimizing the cubic model and attempts to make use of the possible good properties of the cubic step.

For primal-dual interior methods there is an even more compelling reason to combine a line search with cubic regularization. The properties of the barrier function imply that the merit function is undefined outside the feasible region. Because of this, it is common for the trial step to be rejected simply because it generates an iterate that is infeasible for the inequality constraints. In terms of additional matrix factorizations, it can be very expensive to search for a feasible step by repeatedly solving the cubic regularized subproblem for different values of the regularization parameter. Thus, it seems appropriate to use a line search to find a step that remains feasible.

Algorithm 4.1 describes the inner iteration of cubic regularized shifted primal-dual interior-point method combined with Armijo backtracking line-search algorithm.

Algorithm 4.1. Schematic outline of PDARC.

```
1: function PRIMAL-DUAL_INTERIOR-POINT_WITH_CUBIC_REGULARIZATION
2:     Initialization: Given $v_0$, $\mu^B$, $\mu^P$, $\gamma_2 \geq \gamma_1 > 1$, $1 > \eta_2 \geq \eta_1 > 0$, $\mu_0^C > 0$, $1 > \gamma > 0$;
3:     while not converged do
4:         Compute a search direction $\Delta v_k$ from (4.3.1) for which $\mathcal{C}_k(\Delta v_k) \leq \mathcal{C}_k(\Delta v_k^C)$
             where the Cauchy point $\Delta v_k^C = -\gamma_k^C \nabla \mathcal{M}_k$ and $\gamma_k^C = \mathrm{argmin}_{\gamma>0}\, \mathcal{C}_k(-\gamma \nabla \mathcal{M}_k)$;
5:         Set the initial step $\alpha_k \leftarrow 1$;
6:         if not $(s_k + \alpha_k \Delta s_k + \mu^B e > 0$ and $w_k + \alpha_k \Delta w_k > 0)$ then
7:             Find the largest $\alpha_k > 0$ such that $s_k + \alpha_k \Delta s_k + \mu^B e > 0$ and $w_k + \alpha_k \Delta w_k > 0$;
8:         end if
9:         Set $\rho_k \leftarrow \dfrac{\mathcal{M}(v_k + \alpha_k \Delta v_k) - \mathcal{M}(v_k)}{\mathcal{C}_k(\alpha_k \Delta v_k) - \mathcal{M}(v_k)}$;
10:        if $\rho_k \geq \eta_1$ then
11:            Successful iteration: $v_{k+1} \leftarrow v_k + \alpha_k \Delta v_k$;
12:            if $\rho_k \geq \eta_2$ then
13:                Set $\mu_{k+1}^C \in (0, \mu_k^C)$;            [very successful iteration]
14:            else
15:                Set $\mu_{k+1}^C \in [\mu_k^C, \gamma_1 \mu_k^C]$;          [successful iteration]
16:            end if
17:        else
18:            while $\mathcal{M}(v_k + \alpha_k \Delta v_k) - \mathcal{M}(v_k) > \eta_1(\mathcal{C}_k(\alpha_k \Delta v_k) - \mathcal{M}(v_k))$ do
19:                $\alpha_k \leftarrow \gamma \alpha_k$;
20:            end while
21:            $v_{k+1} \leftarrow v_k + \alpha_k \Delta v_k$;
22:            Set $\mu_{k+1}^C \in (\gamma_1 \mu_k^C, \gamma_2 \mu_k^C]$;          [unsuccessful iteration]
23:        end if
24:        Perform a slack reset $s_{k+1} \leftarrow \max\{s_{k+1}, c(x_{k+1}) - \mu^P(y^E + \frac{1}{2}(w_{k+1} - y_{k+1}))\}$;
25:        Set $v_{k+1} \leftarrow (x_{k+1}, s_{k+1}, w_{k+1}, y_{k+1})$;
26:    end while
27: end function
```

In Algorithm 4.1, at the current estimate $v_k = (x_k, s_k, y_k, w_k)$, a step $\Delta v_k = (\Delta x_k, \Delta s_k, \Delta y_k, \Delta w_k)$ is computed as an approximate minimizer that is only required to be at least as good as a suitable Cauchy step of the cubic regularized subproblem

(4.3.1) as described in line 4. Although the Cauchy step is enough for the algorithm to converge, additional conditions on Δv_k may be necessary in order to improve the theoretical properties and performance.

A practical Armijo-type line-search is applied on the search direction $\mathit{\Delta} v_k$ to maintain the feasibility of important quantities and ensure a sufficient decrease on the merit function. In line 7, a step length α_k is computed to keep perturbed slack variables and dual variables positive. Note that compared with the original constraints $c(x) \geq 0$ without slack variables, the steps to the boundary of the constraints $s_k + \alpha_k \Delta s_k + \mu^B e > 0$ and $w_k + \alpha_k \Delta w_k > 0$ can be calculated exactly so that many unnecessary infeasible constraint evaluations can be avoided. In line 9, a ratio ρ_k is computed that measures the agreement between the actual decrease in the objective value of the merit function, $\mathcal{M}(v_k + \alpha_k \Delta v_k) - \mathcal{M}(v_k)$, and the predicted model decrease, $\mathcal{C}_k(\alpha_k \Delta v_k)$. If ρ_k is larger than a threshold η_1, the step is accepted and the next iterate v_{k+1} is set to $v_k + \alpha_k \mathit{\Delta} v_k$. Otherwise, a backtracking line search is performed to find a step length α_k such that the Armijo-type accepting criterion is satisfied in the line 18.

The weight μ_k^C may be viewed as the reciprocal of the trust-region radius. Increasing μ_k^C will result in reducing the size of the search step. If the current weight has produced a good agreement between the actual and the predicted decreases, i.e., a successfully iteration, μ_k^C should be reduced or left unchanged depending on the value of the ratio ρ_k as stated in lines 12–16. If the reduction in merit function is not sufficient, i.e., an unsuccessful iteration, the current weight μ_k^C is increased (see line 22).

The slack-variable reset in the line 24 is designed for efficiency and to handle problems that are locally infeasible, which is analogous to slack-variable resets used in Gill, Murray and Saunders [23], and Gill, Kungurtsev and Robinson [22]. In particular, after reset, the slack variable s_{k+1} satisfies

$$s_{k+1} \geq c(x_{k+1}) - \mu^{P}\left(y^{E} + \frac{1}{2}(w_{k+1} - y_{k+1})\right),$$

which implies, after rearrangement, that

$$c(x_{k+1}) - s_{k+1} \leq \mu^{P}\left(y^{E} + \frac{1}{2}(w_{k+1} - y_{k+1})\right). \tag{4.4.1}$$

The inequality (4.4.1) above guarantees that any limit point (x^{*}, s^{*}) of the sequence $\{(x_k, s_k)\}$ has the property that $c(x^{*}) - s^{*} \leq 0$ if y^{E} and $w_{k+1} - y_{k+1}$ are bounded and μ^{P} converges to zero. This is necessary to handle problems that are locally infeasible, which is a challenge for nonconvex optimization. Moreover, the slack-variable reset never causes the merit function to increase, which implies that the value of the merit function decreases monotonically.

Moreover, we also implement a hybrid approach that uses PDARC and Newton's method. Similar to the algorithm in the section 3.2, we use cubic regularization only on those iterations where we encounter negative curvature. The Newton step is taken when the approximate Hessian B_k is positive definite. As the Newton direction with a suitable step length α is guaranteed to give a sufficient decrease of the merit function, we can simply conduct a line search and find such an α. The details of the hybrid PDARC are summarized in the following Algorithm 4.2.

Algorithm 4.2. Schematic outline of Hybrid PDARC.

```
1: function PRIMAL-DUAL_INTERIOR-POINT_WITH_HYBRID_ARC
2:     Initialization: Given $v_0$, $\mu^B$, $\mu^P$, $\gamma_2 \ge \gamma_1 > 1$, $1 > \eta_2 \ge \eta_1 > 0$, $\mu_0^C > 0$, $1 > \gamma > 0$;
3:     while not converged do
4:         Set cubic_step = true;
5:         if $B(v_k)$ is positive definite then
6:             Compute an approximate solution $\Delta v_k$ of the linear system
                   $$B(v_k)\Delta v_k = -\nabla\mathcal{M}(v_k);$$
7:             Set cubic_step = false;
8:         else
9:             Compute a search direction $\Delta v_k$ from (4.3.1) for which $\mathcal{C}_k(\Delta v_k) \le \mathcal{C}_k(\Delta v_k^C)$
               where the Cauchy point $\Delta v_k^C = -\gamma_k^C \nabla\mathcal{M}_k$ and $\gamma_k^C = \operatorname{argmin}_{\gamma>0} \mathcal{C}_k(-\gamma\nabla\mathcal{M}_k)$;
10:        end if
11:        Set the initial step $\alpha_k \leftarrow 1$;
12:        if not $(s_k + \alpha_k\Delta s_k + \mu^B e > 0$ and $w_k + \alpha_k\Delta w_k > 0)$ then
13:            Find the largest $\alpha_k > 0$ such that $s_k+\alpha_k\Delta s_k+\mu^B e > 0$ and $w_k+\alpha_k\Delta w_k > 0$;
14:        end if
15:        Set $\rho_k \leftarrow \dfrac{\mathcal{M}(v_k + \alpha_k\Delta v_k) - \mathcal{M}(v_k)}{\mathcal{C}_k(\alpha_k\Delta v_k) - \mathcal{M}(v_k)}$;
16:        if $\rho_k \ge \eta_1$ then
17:            Successful iteration: $v_{k+1} \leftarrow v_k + \alpha_k\Delta v_k$;
18:        else
19:            while $\mathcal{M}(v_k + \alpha_k\Delta v_k) - \mathcal{M}(v_k) > \eta_1(\mathcal{C}_k(\alpha_k\Delta v_k) - \mathcal{M}(v_k))$ do
20:                $\alpha_k \leftarrow \gamma\alpha_k$;
21:            end while
22:            $v_{k+1} \leftarrow v_k + \alpha_k\Delta v_k$;
23:        end if
24:        if cubic_step then
25:            Update regularization parameter:
                   $$\mu_{k+1}^C \in \begin{cases} (0, \mu_k^C] & \text{if } \rho_k > \eta_2, & \text{[very successful iteration]} \\ [\mu_k^C, \gamma_1\mu_k^C] & \text{if } \eta_1 \le \rho_k \le \eta_2, & \text{[successful iteration]} \\ [\gamma_1\mu_k^C, \gamma_2\mu_k^C] & \text{otherwise.} & \text{[unsuccessful iteration]} \end{cases}$$
26:        end if
27:        Perform a slack reset $s_{k+1} \leftarrow \max\{s_{k+1}, c(x_{k+1}) - \mu^P(y^E + \frac{1}{2}(w_{k+1} - y_{k+1}))\}$;
28:        Set $v_{k+1} \leftarrow (x_{k+1}, s_{k+1}, w_{k+1}, y_{k+1})$;
29:    end while
30: end function
```

4.5 Theoretical Discussion

In this section, convergence properties of the proposed algorithm are discussed. To prove the convergence of Algorithm 4.1, we first prove some necessary lemmas.

The first result shows that the merit function $\mathcal{M}$ is monotonically decreasing. It is assumed throughout this section that Algorithm 4.1 generates an infinite sequence, i.e., $\nabla\mathcal{M}(v_k) \neq 0$ for all $k \geq 0$.

Lemma 4.5.1. *The sequence of iterates* $\{v_k\}$ *computed by Algorithm* 4.1 *is bounded above and satisfies* $\mathcal{M}(v_{k+1}) < \mathcal{M}(v_k)$ *for all* k.

Proof. From the proof of Lemma 2 in [36] and the assumption that $\nabla\mathcal{M}(v_k)$ is nonzero for all $k \geq 1$, the search direction Δv_k computed from cubic subproblem is a descent direction for $\mathcal{M}$ at v_k, i.e., $\nabla\mathcal{M}\Delta v_k < 0$. This property implies that the line search performed in Algorithm 4.1 produces an α_k such that the new point $v_{k+1} = v_k + \alpha_k \Delta v_k$ satisfies $\mathcal{M}(v_{k+1}) < \mathcal{M}(v_k)$. If follows that the only way the desired result cannot hold is if the slack-reset procedure of step 16 of Algorithm causes $\mathcal{M}$ to increase. The proof is complete if it can be shown that this cannot happen.

It follows that the only way the desired result cannot hold is if the slack-reset procedure of line 24 of Algorithm 4.1 causes $\mathcal{M}$ to increase. The proof is complete if it can be shown that this cannot happen. Let $\widehat{s}_k$ denote the vector

$$\widehat{s}_k = c(x_{k+1}) - \mu^P(y^E + \frac{1}{2}(w_{k+1} - y_{k+1})).$$

The first term of the merit function, $f(x)$, is independent of s, so this term does not change. As the slack-reset procedure has the effect of possibly increasing the value of some of its components, the log barrier terms, $-\sum_{i=1}^{m} \mu^B w_i^E \ln\left(s_i + \mu^B\right)$ and -$\sum_{i=1}^{m} \mu^B w_i^E \ln\left(w_i(s_i + \mu^B)\right)$, can only decrease. We will show the vector $\widehat{s}_k$ used in the slack reset is the unique minimizer of the sum of the rest of terms in the function $\mathcal{M}$, so that the sum of these terms can not increase. Denote the rest part of $\mathcal{M}$ by $\widehat{\mathcal{M}}$ as

$$\widehat{\mathcal{M}}(x,s,y,w) = -\big(c(x)-s\big)^{\mathrm{T}} y^{\scriptscriptstyle E} \\ + \frac{1}{2\mu^{\scriptscriptstyle P}}\|c(x)-s\|^2 + \frac{1}{2\mu^{\scriptscriptstyle P}}\|c(x)-s+\mu^{\scriptscriptstyle P}(y-y^{\scriptscriptstyle E})\|^2 + \sum_{i=1}^{m} w_i(s_i+\mu^{\scriptscriptstyle B}).$$

A simple calculation gives

$$\nabla_s\widehat{\mathcal{M}}(x,s,y,w) = y^{\scriptscriptstyle E} + w + \frac{2}{\mu^{\scriptscriptstyle P}}(s-c(x)-\frac{\mu^{\scriptscriptstyle P}}{2}(y-y^{\scriptscriptstyle E})),$$
$$\nabla_{ss}\widehat{\mathcal{M}}(x,s,y,w) = \frac{2}{\mu^{\scriptscriptstyle P}}I.$$

Setting $\nabla_s\widehat{\mathcal{M}}(x,s,y,w)=0$ gives

$$\widehat{s}_k = c(x) - \mu^{\scriptscriptstyle P}(y^{\scriptscriptstyle E} + \frac{1}{2}(w-y)).$$

As $\mathcal{M}$ is strictly convex with respect to s, it must hold that $\widehat{s}_k$ is the unique minimizer of $\widehat{\mathcal{M}}$. It follows that the slack reset can never increase the value of $\mathcal{M}$, which completes the proof. □

Note that the subproblem (4.3.1) may be converted equivalently to a cubic model with ℓ_2 norm as follows

$$\underset{\Delta v\in\mathbb{R}^{n+3m}}{\text{minimize}}\ \mathcal{M}(v) + \nabla\widehat{\mathcal{M}}(v)\Delta\widehat{v} + \tfrac{1}{2}\Delta\widehat{v}^{\mathrm{T}}\widehat{B}(v)\Delta\widehat{v} + \frac{1}{3}\mu^{\scriptscriptstyle C}\|\Delta\widehat{v}\|_2, \tag{4.5.1}$$

where $\nabla\widehat{\mathcal{M}}(v) = T^{-1/2}\nabla\mathcal{M}(v)$, $\widehat{B}(v) = T^{-1/2}B(v)T^{-1/2}$ and $\Delta\widehat{v} = T^{1/2}\Delta v$. This equivalence allows us to use some useful lemmas established in Section 2.3.

Let $\mathcal{S}$ denote the set of indices of the successful iterations, i.e.,

$$\mathcal{S} = \{k>0 : \mathcal{M}(v_k+\Delta v_k) - \mathcal{M}(v_k) \le \eta_1\left(\mathcal{C}_k(\Delta v_k) - \mathcal{M}(v_k)\right)\}.$$

The following two useful lemmas are proved in [8]. Lemma 4.5.2 provides a guaranteed lower bound on the decrease in the merit function predicted from the cubic model.

Lemma 4.5.2. *Suppose that the step Δv_k satisfies the Cauchy-point condition, i.e., $\mathcal{C}_k(\Delta v_k) \leq \mathcal{C}_k(\Delta v_k^C)$. Then for all $k \geq 0$ we have that*

$$\mathcal{M}(v_k) - \mathcal{C}_k(\Delta v_k) \geq \frac{\|\nabla \mathcal{M}(v_k)\|}{6\sqrt{2}} \min\left\{ \frac{\|\nabla \mathcal{M}(v_k)\|}{1 + \|B(v_k)\|}, \tfrac{1}{2}\sqrt{\frac{\|\nabla \mathcal{M}(v_k)\|}{\mu_k^C}} \right\}.$$

The next lemma yields a useful bound on the step.

Lemma 4.5.3. *Suppose that $\{B(v_k)\}$ is bounded, i.e., $\|B(v_k)\| \leq \kappa_B$ for all $k \geq 0$ and for some constant $\kappa_B > 0$, and that the step Δv_k satisfies the Cauchy-point condition, i.e., $\mathcal{C}_k(\Delta v_k) \leq \mathcal{C}_k(\Delta v_k^C)$. Then*

$$\|\Delta v_k\| \leq 3 \max\left(\frac{\kappa_B}{\mu_k^C}, \sqrt{\frac{\|\nabla \mathcal{M}(v_k)\|}{\mu_k^C}} \right), \quad k \geq 0.$$

Now, we are ready to prove an auxiliary lemma.

Lemma 4.5.4. *Assume $\|B(v_k)\| \leq \kappa_B$ for all $k \geq 0$ and for some constant $\kappa_B > 0$. Suppose that $\mathcal{I}$ is an infinite index set such that $\alpha_k > 0$ and $\|\nabla \mathcal{M}(v_k)\| \geq \epsilon$, for all $k \in \mathcal{I}$ and sine $\epsilon > 0$, and $\sqrt{\frac{\|\nabla \mathcal{M}(v_k)\|}{\mu_k^C}} \to 0$, as $k \to \infty$, $k \in \mathcal{I}$. Then*

$$\|\varDelta v_k\| \leq 3\sqrt{\frac{\|\nabla \mathcal{M}(v_k)\|}{\mu_k^C}} \ \text{ for all } \ k \in \mathcal{I} \ \text{ sufficiently large.} \tag{4.5.2}$$

Additionally, if $v_k \to v^*$, as $k \in \mathcal{I}$, $k \to \infty$ for some $v^* \in \mathbb{R}^{n+3m}$, then each iteration $k \in \mathcal{I}$ that is sufficiently large is very successful, and

$$\mu_{k+1}^C \leq \mu_k^C, \ \text{ for all } \ k \in \mathcal{I} \ \text{ sufficiently large.} \tag{4.5.3}$$

Proof. As $\sqrt{\dfrac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} \to 0$, as $k \to \infty$, $k \in \mathcal{I}$, we have

$$\sqrt{\mu_k^C \|\nabla\mathcal{M}(v_k)\|} = \|\nabla\mathcal{M}(v_k)\| \sqrt{\frac{\mu_k^C}{\|\nabla\mathcal{M}(v_k)\|}} \geq \epsilon \sqrt{\frac{\mu_k^C}{\|\nabla\mathcal{M}(v_k)\|}} \to \infty,$$

for $k \in \mathcal{I}$, $k \to \infty$. By Lemma 4.5.3 and the above inequality, the bound (4.5.2) can be obtained. To prove (4.5.3), we observe that

$$\rho_k \geq \frac{\mathcal{M}(v_k + \alpha_k \Delta v_k) - \mathcal{M}(v_k)}{\mathcal{C}_k(\Delta v_k) - \mathcal{M}(v_k)}.$$

Then, we have the inequality $\rho_k \geq \eta_2$ if

$$r_k \overset{\text{def}}{=} \mathcal{M}(v_k + \alpha_k \mathit{\Delta} v_k) - \mathcal{C}_k(\Delta v_k) + (1 - \eta_2)[\mathcal{C}_k(\Delta v_k) - \mathcal{M}(v_k)] \leq 0. \tag{4.5.4}$$

By a Taylor expansion of $\mathcal{M}(v_k + \alpha_k \mathit{\Delta} v_k)$ centered at v_k, we have for each k

$$\begin{aligned} \mathcal{M}(v_k + \alpha_k \mathit{\Delta} v_k) - \mathcal{C}_k(\Delta v_k) &= \alpha_k (\nabla\mathcal{M}(\xi_k) - \nabla\mathcal{M}(v_k))^{\mathrm{T}} \mathit{\Delta} v_k \\ &\quad - (1 - \alpha_k)\nabla\mathcal{M}(v_k)^{\mathrm{T}} \mathit{\Delta} v_k - \frac{1}{2} \mathit{\Delta} v_k^{\mathrm{T}} B(v_k) \mathit{\Delta} v_k - \frac{\mu_k^C}{3} \|\mathit{\Delta} v\|^3 \\ &\leq \alpha_k (\nabla\mathcal{M}(\xi_k) - \nabla\mathcal{M}(v_k))^{\mathrm{T}} \mathit{\Delta} v_k - (1 - \alpha_k)\nabla\mathcal{M}(v_k)^{\mathrm{T}} \mathit{\Delta} v_k - \frac{1}{2} \mathit{\Delta} v_k^{\mathrm{T}} B(v_k) \mathit{\Delta} v_k, \end{aligned} \tag{4.5.5}$$

for some $\xi_k \in (v_k, v_k + \alpha_k \mathit{\Delta} v_k)$. From (4.5.5) and $\nabla\mathcal{M}(v_k)^{\mathrm{T}} \mathit{\Delta} v_k < 0$, we can obtain

$$\mathcal{M}(v_k + \alpha_k \mathit{\Delta} v_k) - \mathcal{C}_k(\Delta v_k) \leq \alpha_k (\nabla\mathcal{M}(\xi_k) - \nabla\mathcal{M}(x_k))^{\mathrm{T}} \mathit{\Delta} v_k - \tfrac{1}{2} \mathit{\Delta} v_k^{\mathrm{T}} B(v_k) \mathit{\Delta} v_k. \tag{4.5.6}$$

By equation (4.5.2), using (4.5.6) and the assumption $\|B(v_k)\| \leq \kappa_B$, we obtain

$$\begin{aligned}
&\mathcal{M}(v_k + \alpha_k \mathit{\Delta} v_k) - \mathcal{C}_k(\Delta v_k) \\
&\leq 3\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} \left(\alpha_k \|\nabla\mathcal{M}(\xi_k) - \nabla\mathcal{M}(v_k)\| + \frac{3\kappa_B}{2}\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} \right),
\end{aligned} \tag{4.5.7}$$

for all $k \in \mathcal{I}$ sufficiently large. Moreover, by Lemma 4.5.2, together with the assumption $\|\nabla\mathcal{M}(v_k)\| \geq \epsilon$ and the limit $\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} \to 0$, we can bound the remaining term in (4.5.4) as

$$\mathcal{C}_k(\Delta v_k) - \mathcal{M}(v_k) \leq -\frac{\epsilon}{12\sqrt{2}}\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} \tag{4.5.8}$$

for all $k \in \mathcal{I}$ sufficiently large. Then, from equations (4.5.4), (4.5.7) and (4.5.8), it follows

$$r_k \leq 3\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} \left(\alpha_k \|\nabla\mathcal{M}(\xi_k) - \nabla\mathcal{M}(v_k)\| + \frac{3\kappa_B}{2}\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} - \epsilon\frac{1-\eta_2}{12\sqrt{2}} \right), \tag{4.5.9}$$

for all $k \in \mathcal{I}$ sufficiently large. From the assumption $v_k \to v^*$ that $\xi_k \to v^*$, as $k \to \infty$, $k \in \mathcal{I}$, and as $\nabla\mathcal{M}$ is continuous, we can conclude that

$$\|\nabla\mathcal{M}(\xi_k) - \nabla\mathcal{M}(v_k)\| \to 0, k \in \mathcal{I}, k \to \infty.$$

As $\alpha_k \leq 1$ and the limit in (4.5.9) imply that $r_k < 0$ for all $k \in \mathcal{I}$ sufficiently large. Therefore, the inequality (4.5.3) follows from the updating strategy for the weight μ_k^C from the line 12 to the line 16 in Algorithm 4.1. □

We state below the first convergence result for PDARC algorithm. In the following theorem, we show that provided $\mathcal{M}$ is bounded from below, there is a subsequence of $\{\nabla\mathcal{M}(v_k)\}$ converging to zero.

Theorem 4.5.1. *Assume $\|B(v_k)\| \leq \kappa_B$ for all $k \geq 0$ and for some constant $\kappa_B > 0$. If*

$\{\mathcal{M}(v_k)\}$ *is bounded below and* $\mathcal{M}$ *is uniformly continuous on the sequence* $\{v_k\}$*, then*

$$\liminf_{k\to\infty} \|\nabla\mathcal{M}(v_k)\| = 0. \tag{4.5.10}$$

Proof. We will prove this theorem by contradiction. Assume that the result (4.5.10) does not hold, i.e.,

$$\|\nabla\mathcal{M}(v_k)\| \geq \epsilon, \quad \text{for some} \quad \epsilon > 0 \;\; \text{and} \quad k \geq 0. \tag{4.5.11}$$

First, suppose there are only finitely many successful iterations. Denote the index of the last successful iteration as k_0. As all iterations $k \geq k_0 + 1$ are unsuccessful, the weight μ_k^C increases by at least a fraction γ_1 so that

$$\mu_k^C \to \infty \;\; \text{as} \quad k \to \infty. \tag{4.5.12}$$

By Lemma 4.5.3, we can conclude that $v_k \to v^*$ for sufficiently large k, which together with (4.5.11), the second part of Lemma 4.5.4 holds. However, this contradicts with the assumption that all iterations $k \geq k_0 + 1$ are unsuccessful. Therefore $\|\nabla\mathcal{M}(v_k)\| \to 0$ as $k \to \infty$.

Now suppose that there are infinite number of successful iterations. In fact, from the proof of Theorem 2.5 in [8], all sufficiently large iterations belong to $\mathcal{S}$. It follows from Lemma 4.5.2, the assumption (4.5.11) and the construction of the Algorithm 4.1 that

$$\begin{aligned}
\mathcal{M}(v_k) - \mathcal{M}(v_{k+1}) &\geq \eta_1[\mathcal{M}(v_k) - \mathcal{C}_k(\mathit{\Delta} v_k)] \\
&\geq \frac{\eta_1\epsilon}{6\sqrt{2}} \min\left\{\frac{\epsilon}{1+\kappa_B}, \tfrac{1}{2}\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}}\right\} \\
&\geq \frac{\eta_1\epsilon}{12\sqrt{2}}\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}},
\end{aligned} \tag{4.5.13}$$

for all $k \in \mathcal{S}$ sufficiently large, where the last inequality is attained because $\{\mathcal{M}(v_k)\}$ is monotonically decreasing by Lemma 4.5.1 and assumed to be bounded below. For some iteration index k_0 sufficiently large and for any $j \in \mathcal{S}$, $j \geq k_0$, using (4.5.13) and summing up over all sufficiently large iteration gives

$$\mathcal{M}(v_{k_0}) - \mathcal{M}(v_{j+1}) = \sum_{k=k_0, k\in\mathcal{S}}^{j} [\mathcal{M}(v_k) - \mathcal{M}(v_{k+1})] \geq \frac{\eta_1 \epsilon}{12\sqrt{2}} \sum_{k=k_0, k\in\mathcal{S}}^{j} \sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}}. \quad (4.5.14)$$

As $\{\mathcal{M}(v_{j+1})\}$ converges, by letting $j \to \infty$ in (4.5.14), we obtain

$$\sum_{k\in\mathcal{S}} \sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} < +\infty,$$

which further implies

$$\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}} \to 0, \;\text{ as }\; k \to \infty, \;\text{ for }\; k \in \mathcal{S}, \quad (4.5.15)$$

and

$$\mu_k^C \to \infty \;\text{ as }\; k \to \infty. \quad (4.5.16)$$

Hence the first part of Lemma 4.5.4 is established. Then, a simple calculation gives

$$\|v_{l+r} - v_l\| \leq \sum_{k=l}^{l+r-1} \|v_{k+1} - v_k\| = \sum_{k=l}^{l+r-1} \alpha_k \|\Delta v_k\| \leq 3 \sum_{k=l}^{l+r-1} \sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}},$$

for $l \geq 0$ sufficiently large and $r \geq 0$, whose right-hand side tends to zero by taking the limit $l \to \infty$ due to (4.5.15), which yields that $\{v_k\}$ is a Cauchy sequence. and

$$v_k \to v^*, k \to \infty, \;\text{ for some }\; v^* \in \mathbb{R}^{n+3m}. \quad (4.5.17)$$

From (4.5.11), (4.5.15) and (4.5.17), by Lemma 4.5.4, if all $k \in \mathcal{S}$ sufficiently large are very successful, i.e., there are no unsuccessful iteration for k sufficiently large, $\mu_{k+1}^C \leq \mu_k^C$ and

$\{\mu_k^C\}$ is bounded above. This, however, contradicts (4.5.16). Therefore (4.5.11) cannot hold. The proof of the theorem is completed. $\square$

With some additional assumptions, we prove that whole sequence of gradients $\{\nabla\mathcal{M}(v_k)\}$ converges to zero in the following theorem.

Theorem 4.5.2. *Assume $\|B(v_k)\| \leq \kappa_B$ for all $k \geq 0$ and for some constant $\kappa_B > 0$. If $\{\mathcal{M}(v_k)\}$ is bounded below and $\mathcal{M}$ and $\nabla\mathcal{M}$ are uniformly continuous on the sequence $\{v_k\}$, then*

$$\lim_{k\to\infty} \|\nabla\mathcal{M}(v_k)\| = 0. \tag{4.5.18}$$

Proof. If there are only finitely many successful iterations, then the result (4.5.18) follows from the first part of the proof of Theorem 4.5.10. Now suppose that there are an infinite number of successful iterations, i.e., $\mathcal{S}$ is infinite. Assume that there is an infinite subsequence $\{t_i\} \subseteq \mathcal{S}$ such that

$$\|\nabla\mathcal{M}(v_{t_i})\| \geq 2\epsilon, \text{ for some } \epsilon > 0 \text{ and for all } i. \tag{4.5.19}$$

Theorem 4.5.1 implies that for each t_i, there is a first successful iteration $l_i > t_i$ such that $\|\nabla\mathcal{M}(v_{l_i})\| < \epsilon$. Thus $\{l_i\} \subseteq \mathcal{S}$ and for all i, we have

$$\|\nabla\mathcal{M}(v_k)\| \geq \epsilon, \text{ for all } k \text{ with } t_i \leq k < l_i, \text{ and } \|\nabla\mathcal{M}(v_{l_i})\| < \epsilon. \tag{4.5.20}$$

Let $\mathcal{K} \overset{\text{def}}{=} \{k \in \mathcal{S} : t_i \leq k < l_i\}$, where the subsequences $\{t_i\}$ and $\{l_i\}$ are defined above; note that $\mathcal{K}$ is also infinite. It follows from Lemma 4.5.2, the construction of the Algorithm 4.1

and (4.5.20) that

$$\begin{aligned}\mathcal{M}(v_k) - \mathcal{M}(v_{k+1}) &\geq \eta_1[\mathcal{M}(v_k) - \mathcal{C}_k(\varDelta v_k)] \\ &\geq \frac{\eta_1\epsilon}{6\sqrt{2}} \min\left\{\frac{\epsilon}{1+\kappa_B}, \frac{1}{2}\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}}\right\} \\ &\geq \frac{\eta_1\epsilon}{12\sqrt{2}}\sqrt{\frac{\|\nabla\mathcal{M}(v_k)\|}{\mu_k^C}},\end{aligned} \tag{4.5.21}$$

for all $k \in \mathcal{K}$ sufficiently large, where the last inequality is attained because $\{\mathcal{M}(v_k)\}$ is monotonically decreasing by Lemma 4.5.1 and assumed to be bounded below. By Lemma 4.5.3, together with (4.5.21) and the definition of $\mathcal{K}$, we can derive the following bound

$$\mathcal{M}(v_k) - \mathcal{M}(v_{k+1}) \geq \frac{\eta_1\epsilon}{36\sqrt{2}}\|\varDelta v_k\|, \tag{4.5.22}$$

for all $t_i \leq k < l_i$, i sufficiently large. Summing up (4.5.22) over all sufficiently large iteration gives

$$\frac{36\sqrt{2}}{\eta_1\epsilon}[\mathcal{M}(v_{t_i}) - \mathcal{M}(v_{l_i})] \geq \sum_{k=t_i, k\in\mathcal{S}}^{l_i-1} \|v_{k+1} - v_k\| \geq \|v_{t_i} - v_{l_i}\|, \tag{4.5.23}$$

for i sufficiently large. As $\{\mathcal{M}(v_k)\}$ is monotonically decreasing by Lemma 4.5.1 and assumed to be bounded below, the left had side of (4.5.23) converges to zero as $i \to 0$, which implies that $\|v_{t_i} - v_{l_i}\|$ converges to zero as $i \to 0$. As $\nabla\mathcal{M}$ is assumed to be uniformly continuous on the sequence $\{v_k\}$, we have $\|\nabla\mathcal{M}(v_{t_i}) - \nabla\mathcal{M}(v_{l_i})\|$ converges to zero. This, however, contradicts (4.5.19) and (4.5.20). The proof of the theorem is complete. □

4.6 Solving the Cubic Regularized Subproblem

The Algorithm 4.3 for the cubic regularized subproblem (4.3.1) is based on an algorithm of Gertz and Gill [21], which is in turn a modification of the method of Moré and

Sorensen [34] as described in the Section 2.3.3. In this subsection, we omit the subscript k when considering the details associated with a single cubic regularized subproblem.

Recalling the necessary and sufficient conditions in Theorem 4.3.1, we seek for a unique non-negative scalar σ^* such that

$$(B + \sigma^* T)s = -\nabla \mathcal{M}, \quad \|s\|_T - \frac{\sigma^*}{\mu^C} = 0,$$

with $(B + \sigma T)$ positive semidefinite. Let $\psi(\sigma)$ denote the univariate function

$$\psi(\sigma) = \|p_\sigma\|_T - \frac{\sigma}{\mu^C}, \;\text{ where }\; p_\sigma \;\text{ satisfies }\; (B + \sigma T)p_\sigma = -\nabla \mathcal{M}.$$

If σ_{min} is the smallest eigenvalue of $T^{-1/2}BT^{1/2}$, then for any $\sigma > \max(0, -\sigma_{min})$ the matrix $B + \sigma T$ is positive definite and $\psi(\sigma)$ is well-defined. When σ is a positive zero of $\psi(\sigma)$, then $s = p_\sigma$ is a global solution the the cubic subproblem. Moré and Sorensen [34] suggests an safeguarded Newton's method to find an approximate zero of $\psi(\sigma)$. To avoid difficulties associated with the singularities of $\psi(\sigma)$, Gould, Robinson and Thorne [28] uses an alternative approach based on finding a zero of

$$\phi(\sigma; \beta) = \|p_\sigma\|_T^\beta - \left(\frac{\sigma}{\mu^C}\right)^\beta, \;\text{ where }\; \beta \in [-1, \infty) \setminus \{0\}.$$

As in the case of the Moré-Sorensen method, a safeguarded Newton iteration $\mathcal{N}(\sigma)$ is applied with carefully chosen β to generate a nonnegative sequence $\{\sigma_i\}$ and an associated sequence of vectors $\{p_i\}$ such that $B + \sigma T$ is positive definite and $(B + \sigma T)s = -\nabla \mathcal{M}$. The quantity $|\psi(\sigma)|$ is used to measure the accuracy of σ as an approximate zero of $\phi(\sigma; \beta)$.

The Algorithm 4.3 uses the LDL factorization of $K(\sigma)$ to compute the inertia of $B(\sigma)$ with Lemma 4.3.1 in the line 8. This is different from using the Cholesky factorization to check positive-definiteness as proposed by Moré and Sorensen. As $K(\sigma)$ is an $(n+m) \times (n+m)$ symmetric matrix, Algorithm 4.3 has roughly the same computational costs as the algorithm for solving the problem without using any shifts, which is more efficient than factorizing $B(\sigma)$ directly. However, the major drawback is that it is hard to determine a good bound on σ_{min} when $B + \sigma T$ is indefinite. Unlike Cholesky factorization where a direction of negative curvature can be derived in a straightforward way. It is not clear how to obtain such an estimate from the LDL factorization of $K(\sigma)$. However, this does not affect the convergence result.

Another departure from the Moré-Sorensen algorithm is the routine for computing the null vector, $z_{null}(\cdot)$. The routine $z_{null}(\cdot)$ implemented in Moré-Sorensen algorithm is based on the Cholesky factorization of $T^{-1/2}BT^{1/2} + \sigma I$ and the condition number estimator suggested by Cline, Moler, Stewart and Wilkinson [11]. We used a routine that computes an approximate null vector by using Higham's [32] modification of Hager's algorithm [31] supplied with LAPACK in the line 16. This routine is based on estimating the the inverse of the one-norm condition number of the square matrix, where only matrix-vector products with $(T^{-1/2}BT^{1/2} + \sigma I)^{-1}$ are used, rather than matrix factorization. We prefer this routine for practical reason that Hager's algorithm uses inexact arithmetic and avoids the instability associated with the Cholesky factorization of a near-singular matrix.

Algorithm 4.3. Schematic outline of primal-dual cubic subproblem.

```
1: function Primal-Dual_Cubic_Subproblem
2:     Initialization: Given $\sigma_0$, $0 < \sigma_U$, $0 < \epsilon \ll 1$, $0 \ll \theta < 1$;
3:     Estimate the lower bound of the smallest eigenvalue of $B$ as $\bar{\sigma}_{min}$;
4:     Set $\sigma_S \leftarrow \max(0, -\bar{\sigma}_{min})$ and $\sigma_0 \leftarrow \max(\sigma_0, \sigma_S)$;
5:     Set $[\sigma_L,\ \sigma_U] \leftarrow [\sigma_S,\ \max(\sigma_0, \sigma_U)]$;
6:     converged ← false
7:     while not converged do
8:         Compute Inertia of $K(\sigma)$ by LDL factorization: $(\sigma^+, \sigma^-, \sigma^0) \leftarrow \mathrm{In}(K(\sigma_i))$;
9:         if $\sigma^+ < n$ then
10:            $\sigma_{i+1} \leftarrow (1-\theta) * \sigma_L + \theta * \sigma_U$;
11:        else
12:            Compute a vector $p$ by solving the linear system
                   $$B(\sigma_k)p = -\nabla\mathcal{M};$$
13:            if $|\psi(\sigma_i)| \geq \epsilon$ or $\sigma_U - \sigma_L < \epsilon$ then
14:                $s \leftarrow p$; converged ← true;
15:            else if $\psi(\sigma_i) \leq -\epsilon$ then
16:                $z \leftarrow z_{null}(B, \sigma_i, T, p, \mu^C)$;
17:                if $z^{\mathrm{T}}B(\sigma_i)z < \epsilon(2-\epsilon)(p^{\mathrm{T}}B(\sigma_i)p + \sigma_k(\sigma_k/\mu^C)^2)$ then
18:                    $s \leftarrow p + z$; converged ← true;
19:                else
20:                    $\sigma_S = \max(\sigma_S, -z^{\mathrm{T}}B(\sigma_i)z)$;
21:                    $[\sigma_L,\ \sigma_U] = [\max(\sigma_L, \sigma_S),\ \min(\sigma_U, \sigma_i)]$;
22:                end if
23:            end if
24:            if not converged then
25:                $\sigma_N \leftarrow \mathcal{N}(\sigma_i)$;
26:                if $\sigma_N > \sigma_L$ then
27:                    $\sigma_{i+1} \leftarrow \sigma_N$;
28:                else
29:                    $\sigma_{i+1} \leftarrow (1-\theta) * \sigma_L + \theta * \sigma_U$;
30:                end if
31:            end if
32:        end if
33:    end while
34: end function
```

4.7 Practical Considerations for General Case

This section concerns that derivation of the cubic regularization equations for a shifted primal-dual penalty-barrier merit method for constrained optimization. The method are intended for the minimization of a twice-continuously differentiable function subject to both equality and inequality constraints that may include a set of twice-continuously differentiable constraint functions, written in the general form:

$$\underset{x\in\mathbb{R}^n, s\in\mathbb{R}^m}{\text{minimize}} \quad f(x) \quad \text{subject to} \begin{cases} c(x) - s = 0, & L_X s = h_X, \quad \ell^S \le L_L s, \quad L_U s \le u^S, \\ Ax - b = 0, & E_X x = b_X, \quad \ell^X \le E_L x, \quad E_U x \le u^X, \end{cases} \tag{4.7.1}$$

where A denotes a constant $m_A \times n$ matrix, L_X and L_F denote matrices of dimension $m_F \times m$ and $m_X \times m$, respectively, with $m = m_F + m_X$, E_X and E_F are fixed matrices of dimension $n_F \times n$ and $n_X \times n$, respectively, with $n = n_F + n_X$. Throughout the discussion, the functions $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $f : \mathbb{R}^n \mapsto \mathbb{R}$ are assumed to be twice-continuously differentiable. The components of s may be interpreted as slack variables associated with the nonlinear constraints. In addition, it is assumed that a subset of the components of x and s are fixed and that a subset of the other components are subject to upper and lower bounds.

The quantity E_X denotes an $n_X \times n$ matrix formed from n_X independent rows of I_n, the identity matrix of order n. This implies that the equality constraints $E_X x = b_X$ fix n_X components of x at the corresponding values of b_X. Similarly, E_L and E_U denote $n_L \times n$ and $n_U \times n$ matrices formed from subsets of rows of I_n such that $E_X^T E_L = 0$, $E_X^T E_U = 0$, i.e., a variable is either fixed or free to move, possibly bounded by an upper or lower bound. Note that an x_j may be an unrestricted variable in the sense that it is neither fixed nor subject to an upper or lower bound, in which case e_j^T is not a row of E_X, E_L or E_U. Analogous definitions hold for L_X, L_L and L_U as subsets of rows of I_m. However, we impose the restriction that a given s_j must be either fixed or restricted by an upper or lower bound, i.e., there are no unrestricted slacks[1]. Let E_F and L_F denote rows of I_n and I_m such that $\begin{pmatrix} E_X^T & E_F^T \end{pmatrix}$ and $\begin{pmatrix} L_X^T & L_F^T \end{pmatrix}$ are column permutations of I_n and I_m. It follows that the rows of E_L and E_U are a subset of the rows of E_F, and that L_F is formed from the rows of L_L and L_U. These definitions imply that there are $n \times n$ and $m \times m$ permutation matrices P_x and P_s such that

$$P_x = \begin{pmatrix} E_F \\ E_X \end{pmatrix} \quad \text{and} \quad P_s = \begin{pmatrix} L_F \\ L_X \end{pmatrix},$$

with $E_F E_F^T = I_F^x$, $E_X E_X^T = I_X^x$, and $E_F E_X^T = 0$, and $L_F L_F^T = I_F^s$, $L_X L_X^T = I_X^s$, and $L_F L_X^T = 0$.

All general inequality constraints are imposed indirectly through a shifted primal-dual barrier function. The general equality constraints $c(x) - s = 0$ and $Ax = b$ are enforced using an primal-dual augmented Lagrangian algorithm, which implies that the equalities are satisfied in the limit. The exception to this is when the constraints $E_X x = b_X$, and $L_X s = h_X$ are used to fix a subset of the variables and slacks. These bounds are enforced at every iterate. This is intended to allow for the possibility of a variable or slack becoming infeasible with respect to its shifted bound during process of reducing the value of μ^B.

An infeasible slack variable is handled by temporarily fixing it on its bound. An infeasible variable is treated by indirectly enforcing the bound through the use of the primal-dual augmented Lagrangian. Suppose that μ_i^B and $\bar{\mu}_i^B$ denote a shift before and after it is reduced, with $s_i + \mu_i^B > 0$ and $s_i + \bar{\mu}_i^B \leq 0$. The variable s_i can be restored to feasibility by imposing a temporary equality constraint $s_i = 0$. This constraint is

[1]This is not a significant restriction because a "free" slack is equivalent to an unrestricted nonlinear constraint, which may be discarded from the problem. The shifted primal-dual penalty-barrier equations can be derived without this restriction, but the derivation is beyond the scope of this note.

enforced through the primal-dual augmented Lagrangian term until the magnitude of $c_i(x)$ is sufficiently small such that $c_i(x) > -\bar{\mu}_i^B$, at which point s_i is set to $s_i = c_i(x)$ and allowed to vary. If x_j is infeasible with respect to $\ell_j^X - \mu_j^B$, the constraint $x_j - \ell_j^X = 0$ is included as a temporary penalty term in $\mathcal{M}$, i.e.,

$$-v_j^E(x_j - \ell_j^X) + \frac{1}{2\mu_j^A}(x_j - \ell_j^X)^2 + \frac{1}{2\mu_j^A}\big(x_j - \ell_j^X + \mu_j^A(v_j - v_j^E)\big)^2,$$

where v_j^E is an estimate of the multiplier for the constraint $x_j = \ell_j^X$, and μ_j^A is a penalty parameter chosen so that $\mu_j^A < \bar{\mu}_j^B$. The initial values of v_j and v_j^E are $v_j = z_j$ and $v_j^E = z_j^E$, where $z_j > 0$ is the dual variable associated with the constraint $x_j \geq \ell_j^X$. These quantities appear in the perturbed primal-dual optimality conditions associated with problem format (4.7.1). While x_j is infeasible, its associated barrier term is omitted from the shifted primal-dual merit function. Once x_j returns to feasibility for the shifted bound, the shifted barrier term replaces the temporary penalty term in the definition of $\mathcal{M}$ with z_j and z_j^E initialized from v_j and v_j^E. For the purposes of deriving the KKT equations, this scheme implies that additional constraints $Ax - b = 0$ are imposed as in (4.7.1), where A is a matrix of positive and negative rows of I_n and b_j is either ℓ_j^X or $-u_j^X$.

Let x and s be given primal variables and slack variables such that $E_X x = b_X$, $L_X s = h_X$ with $\ell^X - \mu^B < E_L x$, $E_U x < u^X + \mu^B$, $\ell^S - \mu^B < L_L s$, $L_U s < u^S + \mu^B$. Similarly, let z_1, z_2, w_1, w_2 and y denotes dual variables such that $w_1 > 0$, $w_2 > 0$, $z_1 > 0$, and $z_2 > 0$. The partition of x into free and fixed variables induces a partition of H, A, J, E_L and E_U. We use H_F to denote the $n_F \times n_F$ symmetric matrix of rows and columns of H associated with the free variables and A_F, A_X, J_F, J_X to denote the free and fixed columns of A and J. In particular,

$$H_F = E_F H E_F^{\mathrm{T}}, \quad A_F = A E_F^{\mathrm{T}}, \quad A_X = A E_X^{\mathrm{T}}, \quad J_F = J E_F^{\mathrm{T}}, \quad \text{and} \quad J_X = J E_X^{\mathrm{T}},$$

Similarly, the $n_L \times n_F$ matrix E_{LF} and $n_U \times n_F$ matrix E_{UF} comprise the free columns of E_L and E_U, with

$$E_{LF} = E_L E_F^{\mathrm{T}}, \quad \text{and} \quad E_{UF} = E_U E_F^{\mathrm{T}}.$$

It follows that the components of $E_{LF} x_F$ are the values of the free variables that are subject to lower bounds. A similar interpretation applied for $E_{UF} x_F$. Analogous definitions apply for the $m_L \times m_F$ matrix L_{LF} and $m_U \times m_F$ matrix L_{UF}. Consider the diagonal matrices $X_1^\mu = \operatorname{diag}(E_L x - \ell^X + \mu^B e)$, $X_2^\mu = \operatorname{diag}(u^X - E_U x + \mu^B e)$, $Z_1 = \operatorname{diag}(z_1)$, $Z_2 = \operatorname{diag}(z_2)$, $W_1 = \operatorname{diag}(w_1)$, $W_2 = \operatorname{diag}(w_2)$, $S_1^\mu = \operatorname{diag}(L_L s - \ell^S + \mu^B e)$ and $S_2^\mu = \operatorname{diag}(u^S - L_U s + \mu^B e)$. Given the quantities

$$\begin{aligned}
D_Y &= \mu^P I_m, & \pi^Y &= y^E - \frac{1}{\mu^P}(c - s),\\
D_A &= \mu^A I_A, & \pi^V &= v^E - \frac{1}{\mu^A}(Ax - b),\\
(D_1^Z)^{-1} &= (X_1^\mu)^{-1} Z_1, & \pi_1^Z &= \mu^B (X_1^\mu)^{-1} z_1^E,\\
(D_2^Z)^{-1} &= (X_2^\mu)^{-1} Z_2, & \pi_2^Z &= \mu^B (X_2^\mu)^{-1} z_2^E,\\
D_Z &= \left(E_L^{\mathrm{T}} (D_1^Z)^{-1} E_L + E_U^{\mathrm{T}} (D_2^Z)^{-1} E_U\right)^\dagger, & \pi^Z &= E_L^{\mathrm{T}} \pi_1^Z - E_U^{\mathrm{T}} \pi_2^Z,\\
(D_1^W)^{-1} &= (S_1^\mu)^{-1} W_1, & \pi_1^W &= \mu^B (S_1^\mu)^{-1} w_1^E,\\
(D_2^W)^{-1} &= (S_2^\mu)^{-1} W_2, & \pi_2^W &= \mu^B (S_2^\mu)^{-1} w_2^E,\\
D_W &= \left(L_L^{\mathrm{T}} (D_1^W)^{-1} L_L + L_U^{\mathrm{T}} (D_2^W)^{-1} L_U\right)^\dagger, & \pi^W &= L_L^{\mathrm{T}} \pi_1^W - L_U^{\mathrm{T}} \pi_2^W,\\
\breve{D}_W &= \left(D_W^\dagger + \sigma\bar{\sigma} I_F^s\right)^\dagger,
\end{aligned}$$

the KKT system for problem 4.7.1 can be written as follows

$$\begin{pmatrix} H_F(x,y) + \sigma I_F^x + \frac{1}{\bar{\sigma}} A_F^{\mathrm{T}} D_A^{-1} A_F + \frac{1}{\bar{\sigma}} E_F D_Z^{\dagger} E_F^{\mathrm{T}} & -J_F(x)^{\mathrm{T}} \\ J_F(x) & \bar{\sigma}\big(D_Y + \breve{D}_W\big) \end{pmatrix} \begin{pmatrix} \Delta x_F \\ \Delta \widetilde{y} \end{pmatrix}$$
$$= -\begin{pmatrix} E_F\Big(g - J^{\mathrm{T}} y - A^{\mathrm{T}} v - z + \frac{1}{\bar{\sigma}}\big[A^{\mathrm{T}}(v - \pi^V) + z - \pi^Z\big]\Big) \\ D_Y\big(y - \pi^Y\big) + \breve{D}_W\big(\bar{\sigma}(y - w) + w - \pi^W\big) \end{pmatrix}.$$

Then, the solutions of the cubic regularization equations for problem 4.7.1 can be obtained by the following equations

$$\begin{aligned}
\Delta x &= E_F^{\mathrm{T}} \Delta x_F, & \Delta z_1 &= -\frac{1}{1+\sigma}(X_1^{\mu})^{-1}\big(z_1 \cdot (E_L \widehat{x} - \ell^X + \mu^B e) - \mu^B z_1^E\big), \\
\widehat{x} &= x + \Delta x, & \Delta z_2 &= -\frac{1}{1+\sigma}(X_2^{\mu})^{-1}\big(z_2 \cdot (u^X - E_U \widehat{x} + \mu^B e) - \mu^B z_2^E\big), \\
\Delta y &= \Delta \widetilde{y}/(1+2\sigma), & \Delta s &= -\bar{\sigma} \breve{D}_W \Big(y + (1+2\sigma)\Delta y - w + \frac{1}{\bar{\sigma}}\big[w - \pi^W\big]\Big), \\
\widehat{y} &= y + \Delta y, & \Delta w_1 &= -\frac{1}{1+\sigma}(S_1^{\mu})^{-1}\big(w_1 \cdot (L_L \widehat{s} - \ell^S + \mu^B e) - \mu^B w^E 1\big), \\
\widehat{s} &= s + \Delta s, & \Delta w_2 &= -\frac{1}{1+\sigma}(S_2^{\mu})^{-1}\big(w_2 \cdot (u^S - L_U \widehat{s} + \mu^B e) - \mu^B w^E 2\big), \\
\widehat{\pi}^V &= v^E - \frac{1}{\mu^A}(A\widehat{x} - b), & \Delta v &= -\frac{1}{1+\sigma}\big(v - \widehat{\pi}^V\big), \\
w &= L_X^{\mathrm{T}} w_X + L_L^{\mathrm{T}} w_1 - L_U^{\mathrm{T}} w_2, & z &= E_X^{\mathrm{T}} z_X + E_L^{\mathrm{T}} z_1 - E_U^{\mathrm{T}} z_2, \\
\widehat{v} &= v + \Delta v, & \Delta w_X &= [\widehat{y} - w]_X, \\
& & \Delta z_X &= [g + H\Delta x - J^{\mathrm{T}} \widehat{y} - z]_X.
\end{aligned}$$

For further information on the derivation, see Appendix A.

Chapter 5

Numerical Results

In this chapter, we present the numerical results obtained by the algorithms proposed in Chapter 3 for unconstrained optimization and Chapter 4 for constrained optimization. The implementation of these algorithms was done in MATLAB R2019a and all of our experiments were performed on run on a 2018 MacBook Pro with a 2.2 GHz Intel Core i7 and 32 GB of RAM. The numerical performance of the proposed algorithms was evaluated on a set of optimization problems drawn from the Constrained and Unconstrained Testing Environment (CUTEst). The CUTEst test collection is widely used in the optimization community as a standard benchmark for comparing the performance of optimization algorithms. It includes a diverse range of problems, including those from industrial applications, standard academic problems, and problems specifically designed to expose the weaknesses of optimization algorithms. A more detailed description of the CUTEst test collection can be found in the works of Gould et al. [25, 26]. The performance of each algorithm was evaluated based on the number of function evaluations and the number of iterations required to reach a solution within a pre-specified tolerance.

5.1 Performance Profiling

Performance profiling is a useful tool for analyzing the results of numerical experiments on optimization software. The merits of using performance profiles to benchmark optimization software are discussed by Dolan and Moré [16].

The method provides a way to compare the performance of a set of solvers applied to a test set of problems. The main advantage of using performance profiles is that it is intended to standardize the significance of each problem in a set of tests in comparison to the others, and to incorporate data from problems, including those in which one or more solvers were unsuccessful in converging. This is unlike using a basic sum over all converging problems.

The performance profile provides an overview of the performance of a set $\mathcal{S}$ of n_s solvers applied to a test set $\mathcal{P}$ of n_p problems. For each solver $s \in \mathcal{S}$ and problem $p \in \mathcal{P}$ in a profile, the number $t_{p,s}$, which represents the time (or some other measure such as the number of iterations) needed to solve problem p using solver s, is recorded. In order to compare the performance of a problem p across different solvers, the performance ratio $r_{p,s}$ for each successfully solved problem and solver is defined as follows

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in \mathcal{S}\}}.$$

The performance ratio for problems that failed is defined as some value greater than the maximum time needed over all successfully solved problems, denoted by r_M. Note that a solver s is considered to be the best for solving problem p among all solvers in $\mathcal{S}$ if and only if $r_{p,s} = 1$, which means that the solver s has solved the problem p in the shortest amount of time compared to other solvers.

Given the set of performance ratios, Dolan and Moré define a *performance profile* for each solver s as a function

$$\rho_s(\alpha) = \frac{1}{n_p} \left| \{ \, p \in \mathcal{P} : r_{p,s} \le \alpha \, \} \right|,$$

where $\alpha \in [1, r_M]$. If a solver s has the performance profile $\rho_s(\alpha)$, then $\rho_s(\alpha)$ represents the fraction of problems that solver s solves within α times the best solver's performance. The performance profile $\rho_s(\alpha)$ is a monotonically increasing function that is piecewise constant and right continuous. The quantity $\rho_s(1)$ denotes the percentage of problems for which solver s was the most efficient solver among all solvers in the test set. Additionally, $\rho_s(r_M)$ is equal to 1, and $\lim_{\alpha \to r_M^-} \rho_s(\alpha)$ denotes the fraction of problems solved successfully by solver s.

The presented performance profiles are defined as the function

$$\pi_s(\tau) = \frac{1}{n_p} \left| \left\{ p \in \mathcal{P} : \log_2(r_{p,s}) \le \tau \right\} \right| ,$$

with $\tau = \log_2(\alpha)$, which is log-scaled along the horizontal axis to capture behavior near $\alpha = 1$ and $\alpha = r_M$. Specifically, the quantity $\pi_s(\tau)$ can be interpreted as the fraction of problems in the test set that were solved within 2^τ of the best solving time by solver s. In general, the higher and more left the graph $y = \pi_s(\tau)$ is, the better the solver s performs.

5.2 Numerical Results for Unconstrained Optimization

This section compares the numerical performance of the original ARC algorithm [8] and various variants of cubic regularization methods applied to unconstrained optimization problems. The algorithms were implemented using the exact Hessian H_k as B_k in each iteration, along with the exact cubic subproblem solver described in Section 2.3.3. As a benchmark, we also implemented a biased Wolfe trust-region method proposed by Gertz [33].

The numerical experiments were conducted on the unconstrained problems from the CUTEst collection. To ensure computational feasibility within the Matlab environment, we selected testing problems with fewer than 1000 variables, and for those problems whose dimensions could be adjusted, we chose smaller variants. In total, 237 unconstrained problems were included in the test suite.

The stopping criterion for the algorithms was set to be either the gradient norm $\|g_k\|$ less than 10^{-6} or the objective function value f_k less than -10^9. Additionally, the algorithms were allowed to run for a maximum of 3000 iterations or one hour, and any runs exceeding these limits were flagged as failures.

Section 5.2.1 conducts an ablation study and presents performance profiles for the nonmonotone adaptive cubic regularization algorithm (NMARC) introduced in Section 3.1. Section 5.2.2 compares the performance of two variants of the ARC-Newton hybrid method against the vanilla ARC algorithm, NMARC, and the biased Wolfe trust-region method. The performance of the methods is compared using the iteration count and function evaluation count performance profile as the metric. Although it would have been ideal to include a similar figure for CPU times, it was not feasible to do so using the Matlab CPU timer due to its inaccuracy. A more precise comparison of CPU times will be conducted in the future with a carefully implemented version of the methods in FORTRAN or C++, and tested on larger examples in a controlled computing environment.

5.2.1 Numerical Results for Cubic Regularization Methods with Line Search Techniques

In this section, we present a comprehensive numerical comparison of various versions of the nonmonotone adaptive cubic regularization method (NMARC) proposed in Section 3.1, with the vanilla adaptive cubic regularization (ARC) algorithm, and examine the trade-offs between the number of iterations and function evaluations for these algorithms. Specifically, we implemented and evaluated the following five algorithms:

- **ARC**: The vanilla ARC algorithm as stated in Algorithm 2.2 proposed by Cartis, Gould and Toint [8].
- **NMARC-fixed**: The NMARC algorithm as stated in Algorithm 3.1 with a fixed value of 0.8 for the nonmonotonicity parameter β_k.
- **NMARC-adaptive**: The NMARC algorithm as stated in Algorithm 3.1 with an dynamic β_k that depends on the norm of the gradient $\|g_k\|$. The value of β_k is closer to 1 when the iterates are far from the optimum, and closer to 0 when the iterates are near an optimum, which yields better convergence results. Specifically, we define $\beta_k = \min\{0.9, 1 - e^{-\|g_k\|/2}\}$.
- **ARC-Wolfe**: The ARC algorithm combined with monotone Wolfe line search, which is equivalent to the NMARC algorithm with $\beta_k = 0$.
- **ARC-Armijo**: The ARC algorithm combined with Armijo-type backtracking line search.

Table 5.1. Default parameters used in NMARC

Parameter	Value	Parameter	Value
x_0	`prob.x`	η_1	0.01
σ_0	1	η_2	0.9
γ_1	1.05	η_s	0.01
γ_2	3	η_w	0.9

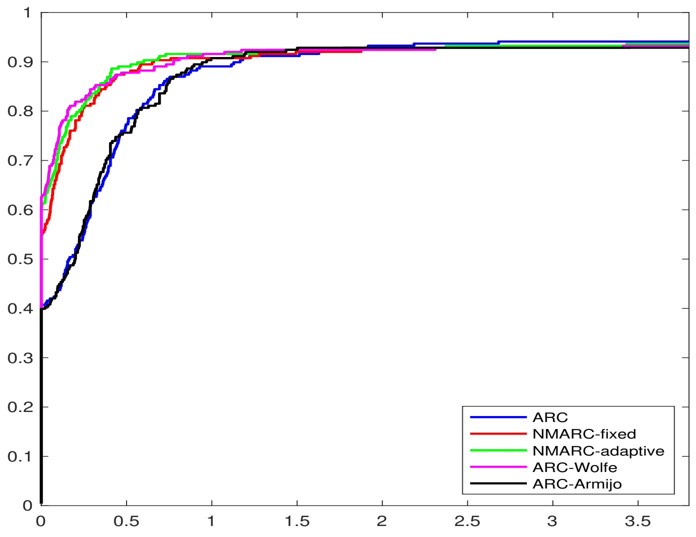

Figure 5.1. Performance profiles of ARC with line search techniques on 237 CUTEst unconstrained problems with respect to the number of iterations.

The MATLAB implementations of the algorithms were initialized with parameter values given in Table 5.2. These values were selected based on their empirical performance on the entire set of problems.

Figure 5.1 and Figure 5.2 give the performance profiles for the total number of iterations and function evaluations. As shown in Figure 5.1, the NMARC algorithms with the Wolfe line search technique generally require fewer iterations than the vanilla ARC algorithm and the ARC algorithm combined with Armijo-type line search. On the other hand, Figure 5.2 suggests that the NMARC algorithm with line search methods typically requires more function evaluations to converge to an optimal solution, which is in line with our expectations. Moreover, based on

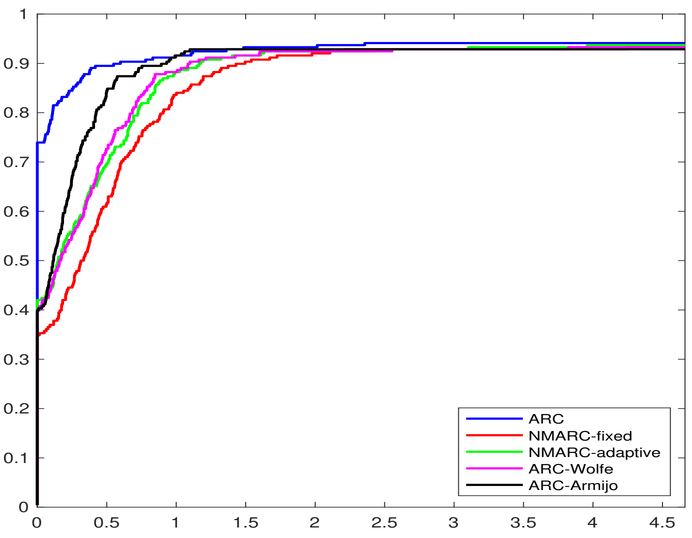

Figure 5.2. Performance profiles of ARC with line search techniques on 237 CUTEst unconstrained problems with respect to the number of function evaluations.

the results presented in Figure 5.1, it can be observed that the NMARC algorithm with both fixed and dynamic nonmonotonicity parameter β_k, as well as the ARC algorithm with Wolfe line search, show similar performance profiles for the number of iterations, which are superior to the performance of the vanilla ARC algorithm and ARC with Armijo-type line search. Regarding the number of function evaluations, the performance profiles in Figure 5.2 indicate that the NMARC algorithm with dynamic nonmonotonicity parameter performs better than NMARC with fixed nonmonotonicity parameter, and has similar performance to the ARC algorithm with monotone Wolfe line search. In terms of the total number of problems solved successfully, out of the 237 test problems, Algorithm ARC achieved optimality on 224 (94.5%) problems. NMARC-fixed and ARC-Wolfe solved 222 (93.7%) problems, while NMARC-adaptive solved 223 (94.1%) problems, and NMARC-Armijo solved 221 (93.2%) problems.

In summary, the numerical results presented in this study support the discussion in Section 3.1 and suggest that NMARC with (monotone or nonmonotone) Wolfe line search is a preferable method when the evaluation of the objective function is not computationally intensive.

5.2.2 Numerical Results for Hybrid Approaches of Cubic Regularization and Newton's Method

Table 5.2. Default parameters used in ARC-Newton

Parameter	Value	Parameter	Value
x_0	`prob.x`	η_1	0.01
σ_0	1	η_2	0.9
γ_1	1.05	η_s	0.01
γ_2	3	η_w	0.9
ϵ_d	10^{-4}		

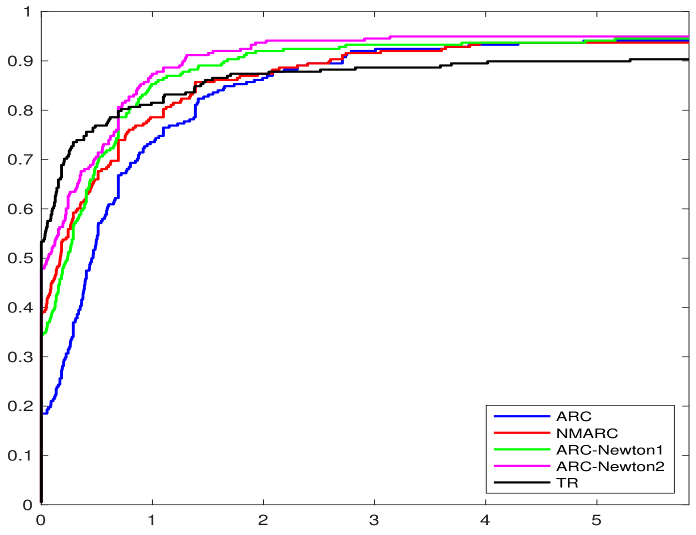

Figure 5.3. Performance profiles of cubic regularization methods on 237 CUTEst unconstrained problems with respect to the number of iterations.

This section presents a performance comparison of hybrid approaches that combine the cubic regularization method with Newton's method, as discussed in Section 3.2, to ARC, NMARC and a biased Wolfe trust-region method. The numerical results demonstrate that the hybrid approach of ARC and Newton's method is a promising strategy for solving nonlinear optimization problems. Specifically, we implemented and evaluated the following five algorithms:

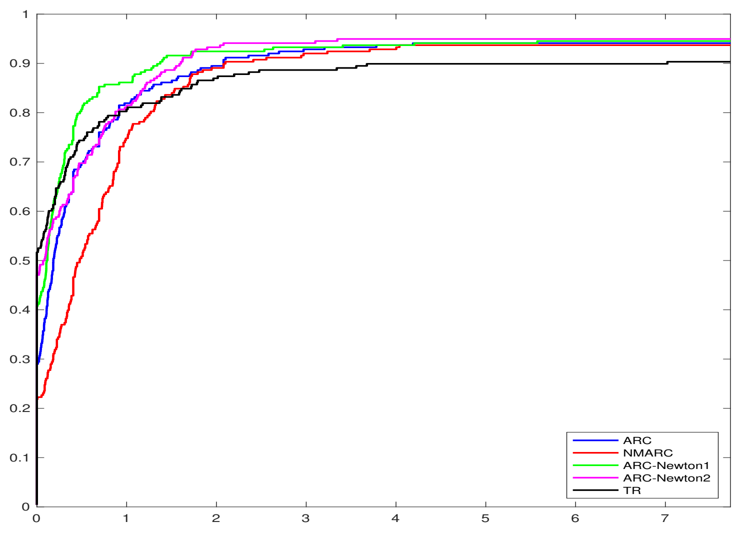

Figure 5.4. Performance profiles of cubic regularization methods on 237 CUTEst unconstrained problems with respect to the number of function evaluations.

- **ARC**: The vanilla ARC algorithm as stated in Algorithm 2.2 proposed by Cartis, Gould and Toint [8].
- **NMARC**: The NMARC algorithm as stated in Algorithm 3.1 with an dynamic β_k that depends on the norm of the gradient $\|g_k\|$. The value of β_k is closer to 1 when the iterates are far from the optimum, and closer to 0 when the iterates are near an optimum, which yields better convergence results. Specifically, we define $\beta_k = \min\{0.9, 1 - e^{-\|g_k\|/2}\}$.
- **ARC-Newton1**: The first hybrid approach of the cubic regularization method and Newton's method outlined in Algorithm 3.2.
- **ARC-Newton2**: The second hybrid approach of the cubic regularization method and Newton's method outlined in Algorithm 3.3.
- **TR**: A trust-region method with a biased Wolfe line search technique proposed by Gertz [33]. To the best of our knowledge this algorithm provides the best numerical performance among trust-region algorithms.

The MATLAB implementations were initialized with parameter values given in Table 5.2.

These parameter values were chosen based on the empirical performance on the entire collection of problems.

Our numerical experiment's performance profiles for the total number of iterations and function evaluations are shown in Figure 5.3 and Figure 5.4, respectively. The performance profiles indicate that while the TR method initially performs well, the hybrid approaches (ARC-Newton1 and ARC-Newton2) eventually outperform the TR method and other variants of cubic regularization algorithm in terms of the total number of problems solved. In particular, problem-by-problem comparisons of the performance profiles indicate that ARC-Newton1 gives the same or fewer number of iterations required for convergence than the vanilla ARC on 85.1% of the problems, and the same or fewer number of function evaluations on 74.3% of the problems. Additionally, it can be observed that ARC-Newton1 and ARC-Newton2 are competitive alternatives to each other. Specifically, while ARC-Newton2 has superior performance in terms of the total number of iterations, ARC-Newton1 outperforms ARC-Newton2 with respect to the total number of function evaluations. Regarding the number of problems solved successfully, out of the 237 test problems, ARC algorithm achieved optimality on 224 (94.5%) problems. NMMAR was successful on 223 (94.1%) problems, while TR solved 214 (90.3%) problems. ARC-Newton1 and ARC-Newton2 achieved even higher success rates, solving 225 (94.9%) and 226 (95.5%) problems, respectively. These results suggest that the hybrid approach of ARC and Newton's method is a significant improvement over vanilla ARC and has the potential to be an effective solution strategy for a wide range of nonlinear optimization problems.

5.3 Numerical Results for Constrained Optimization

This section is devoted to the numerical results of the primal-dual adaptive cubic regularization methods proposed in Chapter 4. We compare the performance of PDARC with that of the benchmark algorithm, a shifted primal-dual penalty-barrier method (PDB) proposed by Gill, Kungurtsev, and Robinson [22]. Specifically, we implemented and evaluated the following three algorithms:

- **PDARC**: The primal-dual adaptive cubic regularization method with Armijo line search technique as described in Algorithm 4.1.
- **PDARC-Hybrid**: The hybrid approach of PDARC algorithm and Newton's method outlined in Algorithm 4.2.
- **PDB**: The shifted primal-dual penalty-barrier method with a specific modified Newton method proposed by Gill, Kungurtsev, and Robinson [22]. This algorithm serves as a benchmark for comparison.

Results were obtained for 349 nonlinear constrained optimization problems from the CUTEst test collection. In order to ensure computational feasibility in the Matlab environment, we limited our test problems to those with a combined total of variables and constraints not exceeding 1000. All the problems selected have at least one constraint (not including any simple upper or lower bounds on variables).

5.3.1 Implementation Details

Each CUTEst problem may be written in the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & \begin{pmatrix} \ell^X \\ \ell^S \end{pmatrix} \le \begin{pmatrix} x \\ c(x) \end{pmatrix} \le \begin{pmatrix} u^X \\ u^S \end{pmatrix}, \end{aligned}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}^m$, $f : \mathbb{R}^n \mapsto \mathbb{R}$, and (ℓ^X, ℓ^S) and (u^X, u^S) are constant vectors of lower and upper bounds. In this format, a fixed variable or an equality constraint has the same value for its upper and lower bound. A variable or constraint with no upper or lower limit is indicated by a bound of $\pm 10^{20}$. In our implementation, each problem was converted to the equivalent form

$$\begin{aligned} \underset{x\in\mathbb{R}^n, s\in\mathbb{R}^m}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) - s = 0, \quad L_X s = h_X, \quad \ell^S \le L_L s, \quad L_U s \le u^S, \\ & \qquad\qquad\qquad\;\; E_X x = b_X, \quad \ell^X \le E_L x, \quad E_U x \le u^X, \end{aligned} \tag{5.3.1}$$

where s is a vector of slack variables. The quantity E_X denotes an $n_X \times n$ matrix formed from n_X independent rows of I_n. Similarly, E_L and E_U denote matrices formed from subsets of I_n such that $E_X^T E_L = 0$, $E_X^T E_U = 0$, i.e., a variable is either fixed or free to move, possibly bounded by an upper or lower bound. Note that a variable x_j need not be subject to a lower or upper bound, or may be bounded below and above, in which case e_j is not a row of E_X, E_L or E_U. Analogous definitions hold for L_X, L_L and L_U as subsets of rows of I_m although a given s_j must be either fixed or restricted by an upper or lower bound, i.e., there are no unrestricted slacks. The bound constraints involving E_X and L_X are enforced explicitly. The problem format (5.3.1) can be extended easily to allow for the possibility of a variable or slack becoming infeasible with respect to its shifted bound. An infeasible slack variable is treated as in the previous section by temporarily fixing it on its bound. An infeasible variable is treated by imposing the bound indirectly using the primal-dual augmented Lagrangian. More details on this can be found in Section 4.7 and Appendix A.

5.3.2 Algorithm Parameters and Termination Conditions

The parameter values used in the MATLAB implementations of PDARC were initialized based on empirical performance on the entire collection of problems. Specifically, the parameter values were chosen according to the tables given in Table 5.3. The primal-dual vector (x_0, y_0) was initialized with default values provided by CUTEst. However, the code immediately projects x_0 onto the feasible region to ensure that it satisfies the bounds on x and is feasible.

Table 5.3. Control parameters and initial values for PDARC and PDARC-Hybrid

Parameter	Value	Parameter	Value
μ_0^C	1.0	η_1	0.01
γ_1	1.05	η_2	0.9
γ_2	3.0	ϵ_d	10^{-6}
$y_{\max}/w_{\max}$	10^6	μ_0^P	1.0
τ_{stop}	10^{-4}	μ_0^B	10^{-3}
τ_0	0.5	$\chi_0^{\max}$	10^3

The iterates were terminated at a point satisfying the condition

$$\|\chi(v_k)\|_\infty < \tau_{\text{stop}}, \tag{5.3.2}$$

where $\chi(v)$ is the optimality measure defined in terms of problem (5.3.1) (see (A.2.1) in Appendix A), or the number of iteration reaches 500. Additionally, if the function value at current iteration is less than -10^9, we consider this problem is unbounded below and terminate the algorithm.

5.3.3 Numerical Results

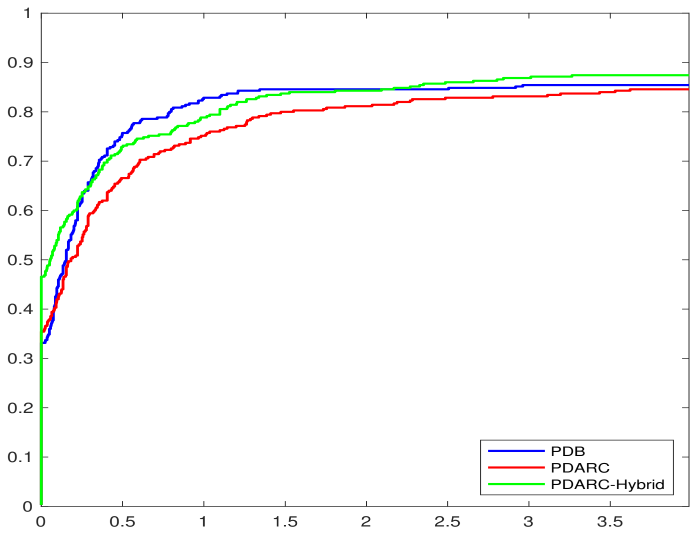

Figure 5.5. Performance profiles on 349 CUTEst constrained problems with respect to the number of iterations.

This section aims to compare the numerical performance of the PDARC, PDARC-Hybrid, and PDB methods. To this end, we present in Figure 5.5 and Figure 5.6 the performance profiles of these algorithms in terms of the total number of iterations and function evaluations. The numerical experiments reveal that the PDARC-Hybrid algorithm provides a competitive alternative to the PDB method. In particular, the performance profile curves of PDARC-Hybrid and PDB exhibit an alternating pattern, with the PDARC-Hybrid curve initially above the PDB

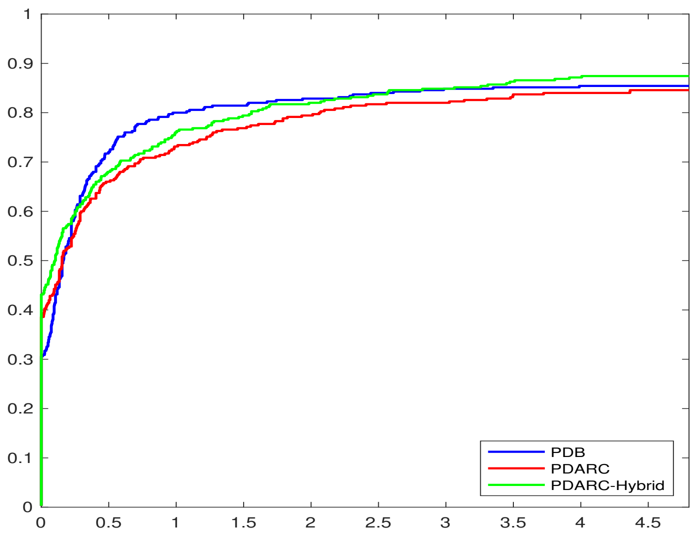

Figure 5.6. Performance profiles on 349 CUTEst constrained problems with respect to the number of function evaluations.

curve, followed by the PDB curve rising above, and ultimately the PDARC-Hybrid curve finishing above the PDB curve. Moreover, problem-by-problem comparisons of the performance profiles indicate that PDARC-Hybrid requires the same or fewer iterations to converge than PDB on 70.2% of the tested problems, and the same or fewer function evaluations on 68.2% of the tested problems. Additionally, it can be observed that PDARC-Hybrid exhibits superior efficiency and reliability compared to PDARC, which is in line with the findings reported for unconstrained problems. Out of the 349 constrained test problems, PDARC was successful on 298 (85.4%) problems, while PDARC-Hybrid achieved optimality on 308 (88.3%) problems, and PDB solved 301 (86.2%) problems. The findings suggest that cubic regularization can be a viable alternative to modified Newton's method when the KKT matrix does not have the correct inertia.

Appendix A

Equations

This section concerns the derivation of the cubic regularization equations for a shifted primal-dual penalty-barrier merit method for constrained optimization. The method are intended for the minimization of a twice-continuously differentiable function subject to both equality and inequality constraints that may include a set of twice-continuously differentiable constraint functions.

A.1 Problem statement

This section concerns the formulation of the equations for problems written in the general form:

$$\underset{x\in\mathbb{R}^n, s\in\mathbb{R}^m}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad \begin{cases} c(x) - s = 0, \;\; L_X s = h_X, \quad \ell^S \le L_L s, \quad L_U s \le u^S, \\ Ax - b = 0, \;\; E_X x = b_X, \quad \ell^X \le E_L x, \quad E_U x \le u^X, \end{cases} \tag{A.1.1}$$

where A denotes a constant $m_A \times n$ matrix, L_X and L_F denote matrices of dimension $m_F \times m$ and $m_X \times m$, respectively, with $m = m_F + m_X$, E_X and E_F are fixed matrices of dimension $n_F \times n$ and $n_X \times n$, respectively, with $n = n_F + n_X$. Throughout the discussion, the functions $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $f : \mathbb{R}^n \mapsto \mathbb{R}$ are assumed to be twice-continuously differentiable. The components of s may be interpreted as slack variables associated with the nonlinear constraints. In addition, it is assumed that a subset of the components of x and s are fixed and that a subset of the other components are subject to upper and lower bounds.

The quantity E_X denotes an $n_X \times n$ matrix formed from n_X independent rows of I_n, the identity matrix of order n. This implies that the equality constraints $E_X x = b_X$ fix n_X components of x at the corresponding values of b_X. Similarly, E_L and E_U denote $n_L \times n$ and $n_U \times n$ matrices formed from subsets of rows of I_n such that $E_X^T E_L = 0$, $E_X^T E_U = 0$, i.e., a variable is either fixed or free to move, possibly bounded by an upper or lower bound. Note that an x_j may be an unrestricted variable in the sense that it is neither fixed nor subject to an upper or lower bound, in which case e_j^T is not a row of E_X, E_L or E_U. Analogous definitions hold for L_X, L_L and L_U as subsets of rows of I_m. However, we impose the restriction that a given s_j must be either fixed or restricted by an upper or lower bound, i.e., there are no unrestricted slacks[1]. Let E_F and L_F denote rows of I_n and I_m such that $\begin{pmatrix} E_X^T & E_F^T \end{pmatrix}$ and $\begin{pmatrix} L_X^T & L_F^T \end{pmatrix}$ are column permutations of I_n and I_m. It follows that the rows of E_L and E_U are a subset of the rows of E_F, and that L_F is formed from the rows of L_L and L_U. These definitions imply that there are $n \times n$ and $m \times m$ permutation matrices P_x and P_s such that

$$P_x = \begin{pmatrix} E_F \\ E_X \end{pmatrix} \quad \text{and} \quad P_s = \begin{pmatrix} L_F \\ L_X \end{pmatrix}, \tag{A.1.2}$$

with $E_F E_F^T = I_F^x$, $E_X E_X^T = I_X^x$, and $E_F E_X^T = 0$, and $L_F L_F^T = I_F^s$, $L_X L_X^T = I_X^s$, and $L_F L_X^T = 0$.

All general inequality constraints are imposed indirectly using a shifted primal-dual barrier function. The general equality constraints $c(x) - s = 0$ and $Ax = b$ are enforced using an primal-dual augmented Lagrangian algorithm, which implies that the

[1]This is not a significant restriction because a "free" slack is equivalent to a unrestricted nonlinear constraint, which may be discarded from the problem. The shifted primal-dual penalty-barrier equations can be derived without this restriction, but the derivation is beyond the scope of this note.

equalities are satisfied in the limit. The exception to this is when the constraints $E_X x = b_X$, and $L_X s = h_X$ are used to fix a subset of the variables and slacks. These bounds are enforced at every iterate.

An equality constraint $c_i(x) = 0$ may be handled by introducing the slack variable s_i and writing the constraint as the two constraints $c_i(x) - s_i = 0$ and $s_i = 0$. In this case the ith coordinate vector e_i can be included as a row of L_X. Linear *inequality* constraints must be included as part of c. A linear equality constraint can be either included with the nonlinear equality constraints or the matrix A. The constraints involving A may be used to temporarily fix a subset of the variables at their bounds without altering the underlying structure of the approximate Newton equations. In this case, the associated rows of A are rows of the identity matrix.

The optimality conditions for problem (A.1.1) are given in Section A.2. The shifted path-following equations are formulated in Section A.3. The shifted primal-dual penalty-barrier function associated with problem is discussed in Section A.4. This function serves as a merit function for both the cubic regularization method. The equations for a cubic regularized Newton method are formulated in Sections A.6, and summarized in Section A.7.

Notation.

Given vectors x and y, the vector consisting of x augmented by y is denoted by (x, y). The subscript i is appended to vectors to denote the ith component of that vector, whereas the subscript k is appended to a vector to denote its value during the kth iteration of an algorithm, e.g., x_k represents the value for x during the kth iteration, whereas $[x_k]_i$ denotes the ith component of the vector x_k. Given vectors a and b with the same dimension, the vector with ith component $a_i b_i$ is denoted by $a \cdot b$. Similarly, $\min(a, b)$ is a vector with components $\min(a_i, b_i)$. The vector e denotes the column vector of ones, and I denotes the identity matrix. The dimensions of e and I are defined by the context. The vector two-norm or its induced matrix norm are denoted by $\|\cdot\|$. The vector $\nabla f(x)$ is used to denote the gradient of $f(x)$. The matrix $J(x)$ denotes the $m \times n$ constraint Jacobian, which has ith row $\nabla c_i(x)^T$. Given a Lagrangian function $L(x, y) = f(x) - c(x)^{\mathrm{T}} y$ with y a m-vector of dual variables, the Hessian of the Lagrangian with respect to x is denoted by

$H(x,y) = \nabla^2 f(x) - \sum_{i=1}^m y_i \nabla^2 c_i(x)$. The cubic regularization equations utilize the Moore-Penrose pseudoinverse of a diagonal matrix. In particular, if $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$, then the pseudoinverse $D^\dagger$ is diagonal with $D^\dagger_{ii} = 0$ for $d_i = 0$ and $D^\dagger_{ii} = 1/d_i$ for $d_i \neq 0$.

A.2 Optimality conditions

The first-order KKT conditions for problem (A.1.1) are

$$\left.\begin{array}{rrr}
\nabla f(x^*) - J(x^*)^{\mathrm{T}} y^* - A^{\mathrm{T}} v^* - E_X^T z_X^* - E_L^T z_1^* + E_U^T z_2^* = 0, & z_1^* \geq 0, & z_2^* \geq 0, \\
y^* - L_X^T w_X^* - L_L^T w_1^* + L_U^T w_2^* = 0, & w_1^* \geq 0, & w_2^* \geq 0, \\
c(x^*) - s^* = 0, & E_X x^* - b_X = 0, & L_X s^* - h_X = 0, \\
A x^* - b = 0, & & \\
E_L x^* - \ell^X \geq 0, & u^X - E_U x^* \geq 0, & \\
L_L s^* - \ell^S \geq 0, & u^S - L_U s^* \geq 0, & \\
z_1^* \cdot (E_L x^* - \ell^X) = 0, & z_2^* \cdot (u^X - E_U x^*) = 0, & \\
w_1^* \cdot (L_L s^* - \ell^S) = 0, & w_2^* \cdot (u^S - L_U s^*) = 0, &
\end{array}\right\} \qquad \text{(A.2.1)}$$

where y^*, w_X^*, and z_X^* are the multipliers for the equality constraints $c(x) - s = 0$, $L_X s^* = h_X$ and $E_X x^* = b_X$, and w_1^*, w_2^*, z_1^* and z_2^*, and may be interpreted as the Lagrange multipliers for the inequality constraints $L_L s - \ell^S \geq 0$, $u^S - L_U s \geq 0$, $E_L x - \ell^X \geq 0$ and $u^X - E_U x \geq 0$ respectively. The components of v^* are the multipliers for the linear equality constraints $Ax = b$. If $x_1 = E_L x - \ell^X$, $x_2 = u^X - E_U x$, $s_1 = L_L s - \ell^S$, and $s_2 = u^S - L_U s$, then z_1^*, z_2^*, w_1^*, and w_2^* are the Lagrange multipliers for the inequality constraints

$x_1 \geq 0$, $x_2 \geq 0$, $s_1 \geq 0$, and $s_2 \geq 0$, respectively. In the derivations that follow, the vectors z and w are defined as

$$z = E_X^T z_X + E_L^T z_1 - E_U^T z_2, \quad \text{and} \quad w = L_X^T w_X + L_L^T w_1 - L_U^T w_2. \tag{A.2.2}$$

A.3 The path-following equations

Let z^E1 and z^E2, w^E1 and w^E2 denote nonnegative estimates of z_1^* and z_2^*, w_1^* and w_2^*. Given small positive scalars μ^P, μ^A and μ^B, consider the perturbed optimality conditions

$$\left.\begin{array}{rrr}
\nabla f(x) - J(x)^{\mathrm{T}} y - A^{\mathrm{T}} v - E_X^T z_X - E_L^T z_1 + E_U^T z_2 = 0, & z_1 \geq 0, & z_2 \geq 0, \\
y - L_X^T w_X - L_L^T w_1 + L_U^T w_2 = 0, & w_1 \geq 0, & w_2 \geq 0, \\
c(x) - s = \mu^P (y^E - y), & E_X x^* - b_X = 0, & L_X s^* - h_X = 0, \\
Ax - b = \mu^A (v^E - v), & & \\
E_L x - \ell^X \geq 0, & u^X - E_U x \geq 0, & \\
L_L s - \ell^S \geq 0, & u^S - L_U s \geq 0, & \\
z_1 \cdot (E_L x - \ell^X) = \mu^B (z^E 1 - z_1), & z_2 \cdot (u^X - E_U x) = \mu^B (z^E 2 - z_2), & \\
w_1 \cdot (L_L s - \ell^S) = \mu^B (w^E 1 - w_1), & w_2 \cdot (u^S - L_U s) = \mu^B (w^E 2 - w_2). &
\end{array}\right\} \tag{A.3.1}$$

Let v_P denote the vector of variables $v_P = (x,\ s,\ y,\ v,\ w_X,\ z_X,\ z_1,\ z_2,\ w_1,\ w_2)$. The primal-dual path-following equations are given by $F(v_P)$, with

$$F(v_P) = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y - A^{\mathrm{T}} v - E_X^T z_X - E_L^T z_1 + E_U^T z_2 \\ y - L_X^T w_X - L_L^T w_1 + L_U^T w_2 \\ c(x) - s + \mu^P (y - y^E) \\ Ax - b + \mu^A (v - v^E) \\ E_X x - b_X \\ L_X s - h_X \\ z_1 \cdot (E_L x - \ell^X) + \mu^B (z_1 - z^E 1) \\ z_2 \cdot (u^X - E_U x) + \mu^B (z_2 - z^E 2) \\ w_1 \cdot (L_L s - \ell^S) + \mu^B (w_1 - w^E 1) \\ w_2 \cdot (u^S - L_U s) + \mu^B (w_2 - w^E 2) \end{pmatrix} = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y - A^{\mathrm{T}} v - z \\ y - w \\ c(x) - s + \mu^P (y - y^E) \\ Ax - b + \mu^A (v - v^E) \\ E_X x - b_X \\ L_X s - h_X \\ z_1 \cdot (E_L x - \ell^X) + \mu^B (z_1 - z^E 1) \\ z_2 \cdot (u^X - E_U x) + \mu^B (z_2 - z^E 2) \\ w_1 \cdot (L_L s - \ell^S) + \mu^B (w_1 - w^E 1) \\ w_2 \cdot (u^S - L_U s) + \mu^B (w_2 - w^E 2) \end{pmatrix}. \tag{A.3.2}$$

(To simplify the notation, the dependence of F on the parameters μ^A, μ^P, μ^B, y^E, v^E, z^E1, z^E2, w^E1, w^E2 is omitted.) Any zero $(x,\ s,\ y,\ v,\ w_X,\ z_X,\ z_1,\ z_2,\ w_1,\ w_2)$ of F such that $\ell^X < E_L$, $E_U x < u^X$, $\ell^S < L_L s$, $L_U < u^S$, $z_1 > 0$, $z_2 > 0$, $w_1 > 0$, and $w_2 > 0$ approximates a point satisfying the optimality conditions (A.2.1), with the approximation becoming increasingly accurate as the terms $\mu^P(y - y^E)$, $\mu^A(v - v^E)$, $\mu^B(z_1 - z^E 1)$, $\mu^B(z_2 - z^E 2)$, $\mu^B(w_1 - w^E 1)$ and $\mu^B(w_2 - w^E 2)$ approach zero. For any sequence of z^E1, z^E2, w^E1, w^E2, v^E and y^E such that $z^E1 \to z_1^*$, $z^E2 \to z_2^*$, $w^E1 \to w_1^*$, $w^E2 \to w_2^*$, $v^E \to v^*$ and $y^E \to y^*$, and it must hold that solutions $(x, s, y, v, z_1, z_2, w_1, w_2)$ of (A.3.1) must satisfy $z_1 \cdot (x - \ell^X) \to 0$, $z_2 \cdot (u^X - x) \to 0$, $w_1 \cdot (s - \ell^S) \to 0$, and $w_2 \cdot (u^S - s) \to 0$, This implies that any solution $(x,\ s,\ y,\ v,\ w_X,\ z_X,\ z_1,\ z_2,\ w_1,\ w_2)$ of (A.3.1) will approximate a solution of (A.2.1) independently of the values of μ^P, μ^A and μ^B (i.e., it is not necessary that $\mu^P \to 0$, $\mu^A \to 0$ and $\mu^B \to 0$).

If $v_P = (x,\ s,\ y,\ v,\ w_X,\ z_X,\ z_1,\ z_2,\ w_1,\ w_2)$ is a given approximate zero of F such that $\ell^X - \mu^B < E_L x$, $E_U x < u^X + \mu^B$, $\ell^S - \mu^B < L_L s$, $L_U s < u^S + \mu^B$, $z_1 > 0$, $z_2 > 0$, $w_1 > 0$, and $w_2 > 0$, the Newton equations for the change in variables $\Delta v_P = (\Delta x,\ \Delta s,\ \Delta y,\ \Delta v,\ \Delta w_X,\ \Delta z_X,\ \Delta z_1,\ \Delta z_2,\ \Delta w_1,\ \Delta w_2)$ are given by $F'(v_P)\Delta v_P = -F(v_P)$, with

$$F'(v_P) = \begin{pmatrix} H & 0 & -J^T & -A^T & 0 & -E_X^T & -E_L^T & E_U^T & 0 & 0 \\ 0 & 0 & I_m & 0 & -L_X^T & 0 & 0 & 0 & -L_L^T & L_U^T \\ J & -I_m & D_Y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & D_A & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_1 E_L & 0 & 0 & 0 & 0 & 0 & X_1^\mu & 0 & 0 & 0 \\ -Z_2 E_U & 0 & 0 & 0 & 0 & 0 & 0 & X_2^\mu & 0 & 0 \\ 0 & W_1 L_L & 0 & 0 & 0 & 0 & 0 & 0 & S^\mu 1 & 0 \\ 0 & -W_2 L_U & 0 & 0 & 0 & 0 & 0 & 0 & 0 & S^\mu 2 \end{pmatrix} \qquad \text{(A.3.3)}$$

(recall that $z = E_X^T z_X + E_L^T z_1 - E_U^T z_2$ and $w = L_X^T w_X + L_L^T w_1 - L_U^T w_2$). Any s may be written as $s = L_F^T s_F + L_X^T s_X$, where s_F and s_X denote the components of s corresponding to the “free” and “fixed” components of s, respectively. Similarly, any x may be written as $x = E_F^T x_F + E_X^T x_X$, where x_F and x_X denote the free and fixed components of x.

The partition of x into free and fixed variables induces a partition of H, A, J, E_L and E_U. We use H_F to denote the $n_F \times n_F$ symmetric matrix of rows and columns of H associated with the free variables and A_F, A_X, J_F, J_X to denote the free and fixed columns of A and J. In particular,

$$H_F = E_F H E_F^T, \qquad A_F = AE_F^T, \qquad A_X = AE_X^T, \qquad J_F = JE_F^T, \quad \text{and} \quad J_X = JE_X^T,$$

Similarly, the $n_L \times n_F$ matrix E_{LF} and $n_U \times n_F$ matrix E_{UF} comprise the free columns of E_L and E_U, with

$$E_{LF} = E_L E_F^T, \quad \text{and} \quad E_{UF} = E_U E_F^T.$$

It follows that the components of $E_{LF}x_F$ are the values of the free variables that are subject to lower bounds. A similar interpretation applied for $E_{UF}x_F$. Analogous definitions apply for the $m_L \times m_F$ matrix L_{LF} and $m_U \times m_F$ matrix L_{UF}.

The next step is to transform the path-following equations to reflect the structure of free and fixed variables. Consider the block-diagonal orthogonal matrix $Q = \text{diag}(P_X,\, P_S,\, I_m,\, I_A,\, I^s_X,\, I^x_X,\, I^x_L,\, I^x_U,\, I^s_L,\, I^s_U)$, where P_X and P_S are defined in (A.1.2). Given the identities

$$\begin{pmatrix} \Delta x_F \\ \Delta x_X \end{pmatrix} = P_x \Delta x = \begin{pmatrix} E_F \Delta x \\ E_X \Delta x \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \Delta s_F \\ \Delta s_X \end{pmatrix} = P_S \Delta s = \begin{pmatrix} L_F \Delta s \\ L_X \Delta s \end{pmatrix},$$

and $QF'(v_P)Q^{\mathrm{T}}Q\Delta v_P = QF(v_P)$, we obtain the transformed equations

$$\begin{pmatrix} H_F & H_O^T & 0 & 0 & -J_F^T & -A_F^T & 0 & 0 & -E_{LF}^T & E_{UF}^T & 0 & 0 \\ H_O & H_X & 0 & 0 & -J_X^T & -A_X^T & 0 & -I_X^x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L_F & 0 & 0 & 0 & 0 & 0 & -L_{LF}^T & L_{UF}^T \\ 0 & 0 & 0 & 0 & L_X & 0 & -I_X^s & 0 & 0 & 0 & 0 & 0 \\ J_F & J_X & -L_F^T & -L_X^T & D_Y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_F & A_X & 0 & 0 & 0 & D_A & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_X^s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_X^x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_1E_{LF} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_1^\mu & 0 & 0 & 0 \\ -Z_2E_{UF} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_2^\mu & 0 & 0 \\ 0 & 0 & W_1L_{LF} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & S^\mu 1 & 0 \\ 0 & 0 & -W_2L_{UF} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & S^\mu 2 \end{pmatrix} \begin{pmatrix} \Delta x_F \\ \Delta x_X \\ \Delta s_F \\ \Delta s_X \\ \Delta y \\ \Delta v \\ \Delta w_X \\ \Delta z_X \\ \Delta z_1 \\ \Delta z_2 \\ \Delta w_1 \\ \Delta w_2 \end{pmatrix}$$

$$= \begin{pmatrix} g_F - J_F^{\mathrm{T}} y - A_F^{\mathrm{T}} v - z_F \\ g_X - J_X^T y - A_X^T v - z_X \\ y_F - w_F \\ y_X - w_X \\ c(x) - s + \mu^P (y - y^E) \\ Ax - b + \mu^A (v - v^E) \\ E_X x - b_X \\ L_X s - h_X \\ z_1 \cdot (E_L x - \ell^X) + \mu^B (z_1 - z^E 1) \\ z_2 \cdot (u^X - E_U x) + \mu^B (z_2 - z^E 2) \\ w_1 \cdot (L_L s - \ell^S) + \mu^B (w_1 - w^E 1) \\ w_2 \cdot (u^S - L_U s) + \mu^B (w_2 - w^E 2) \end{pmatrix},$$

where $g_F = E_F g$, $z_F = E_F z$ and $y_F = L_F y$.

As the constraints $L_X s - h_X = 0$ and $E_X x - b_X = 0$ are enforced throughout, it follows that $\Delta s_X = 0$ and $\Delta x_X = 0$, in which case Δs and Δx satisfy

$$\Delta s = L_F^T \Delta s_F + L_X^T \Delta s_X = L_F^T \Delta s_F \quad \text{and} \quad \Delta x = E_F^T \Delta x_F + E_X^T \Delta x_X = E_F^T \Delta x_F.$$

After scaling the last four blocks of equations by (respectively) Z_1^{-1}, Z_2^{-1}, W_1^{-1} and W_2^{-1}, collecting terms and reordering the equations and unknowns, we obtain

$$\begin{pmatrix} H_F & 0 & -J_F^T & -A_F^T & -E_{LF}^T & E_{UF}^T & 0 & 0 \\ 0 & 0 & L_F & 0 & 0 & 0 & -L_{LF}^T & L_{UF}^T \\ J_F & -L_F^T & D_Y & 0 & 0 & 0 & 0 & 0 \\ A_F & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & D_Z1 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & D_Z2 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & D_W1 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & D_W2 \end{pmatrix} \begin{pmatrix} \Delta x_F \\ \Delta s_F \\ \Delta y \\ \Delta v \\ \Delta z_1 \\ \Delta z_2 \\ \Delta w_1 \\ \Delta w_2 \end{pmatrix} = - \begin{pmatrix} g_F - J_F^{\mathrm{T}} y - A_F^T v - z_F \\ y_F - w_F \\ -D_Y(\pi^Y - y) \\ -D_A(\pi^V - v) \\ -D_Z1(\pi_1^Z - z_1) \\ -D_Z2(\pi_2^Z - z_2) \\ -D_W1(\pi_1^W - w_1) \\ -D_W2(\pi_2^W - w_2) \end{pmatrix}, \tag{A.3.4}$$

where

$$\left.\begin{array}{llll} D_Y = \mu^P I_m, & \pi^Y = y^E - \dfrac{1}{\mu^P}(c - s), & D_A = \mu^A I_A, & \pi^V = v^E - \dfrac{1}{\mu^A}(Ax - b), \\ D_W1 = S^\mu 1 W_1^{-1}, & \pi_1^W = \mu^B (S^\mu 1)^{-1} w^E 1, & D_Z1 = X_1^\mu Z_1^{-1}, & \pi_1^Z = \mu^B (X_1^\mu)^{-1} z^E 1, \\ D_W2 = S^\mu 2 W_2^{-1}, & \pi_2^W = \mu^B (S^\mu 2)^{-1} w^E 2, & D_Z2 = X_2^\mu Z_2^{-1}, & \pi_2^Z = \mu^B (X_2^\mu)^{-1} z^E 2. \end{array}\right\} \tag{A.3.5}$$

Given the definitions (A.2.2), the vectors Δs and Δw_X are recovered as $\Delta s = L_F^T \Delta s_F$ and $\Delta w_X = [y + \Delta y - w]_X$. Similarly, Δx and Δz_X are recovered as $\Delta x = E_F^T \Delta x_F$ and $\Delta z_X = [g + H\Delta x - J^{\mathrm{T}}(y + \Delta y) - z]_X$.

A.4 A shifted primal-dual penalty-barrier function

Problem (A.1.1) is equivalent to

$$\begin{array}{ll} \underset{x,x_1,x_2,s,s_1,s_2}{\text{minimize}} & f(x) \\ \text{subject to} & c(x) - s = 0, \qquad Ax - b = 0, \\ & E_L x - x_1 = \ell^X, \quad L_L s - s_1 = \ell^S, \quad x_1 \geq 0, \quad s_1 \geq 0, \\ & E_U x + x_2 = u^X, \quad L_U s + s_2 = u^S, \quad x_2 \geq 0, \quad s_2 \geq 0, \\ & E_X x - b_X = 0, \quad L_X s - h_X = 0. \end{array}$$

Consider the shifted primal-dual penalty-barrier problem

$$\left.\begin{array}{ll} \underset{x,x_1,x_2,s,s_1,s_2,y,v,z_1,z_2,w_1,w_2}{\text{minimize}} & M(x, x_1, x_2, s, s_1, s_2, y, v, w_1, w_2\,; \mu^P, \mu^B, y^E, v^E, w^E1, w^E2) \\ \text{subject to} & E_L x - x_1 = \ell^X, \quad L_L s - s_1 = \ell^S, \; x_1 + \mu^B e > 0, \; z_1 > 0, \; s_1 + \mu^B e > 0, \; w_1 > 0, \\ & E_U x + x_2 = u^X, \quad L_U s + s_2 = u^S, \; x_2 + \mu^B e > 0, \; z_2 > 0, \; s_2 + \mu^B e > 0, \; w_2 > 0, \\ & E_X x - b_X = 0, \quad L_X s - h_X = 0, \end{array}\right\} \qquad \text{(A.4.1)}$$

where $M(x, x_1, x_2, s, s_1, s_2, y, v, z_1, z_2, w_1, w_2\,; \mu^P, \mu^B, y^E, v^E, z^E 1, z^E 2, w^E 1, w^E 2)$ is the shifted primal-dual penalty-barrier function

$$\begin{aligned} f(x) - \big(c(x) - s\big)^{\mathrm{T}} y^E + \frac{1}{2\mu^P}\|c(x) - s\|^2 + \frac{1}{2\mu^P}\|c(x) - s + \mu^P(y - y^E)\|^2 \\ - (Ax - b)^{\mathrm{T}} v^E + \frac{1}{2\mu^A}\|Ax - b\|^2 + \frac{1}{2\mu^A}\|Ax - b + \mu^A(v - v^E)\|^2 \\ - \sum_{j=1}^{n_F} \Big\{ \mu^B [z^E 1]_j \ln\big([z_1]_j [x_1 + \mu^B e]_j^2\big) - [z_1 \cdot (x_1 + \mu^B e)]_j \Big\} \\ - \sum_{j=1}^{n_F} \Big\{ \mu^B [z^E 2]_j \ln\big([z_2]_j [x_2 + \mu^B e]_j^2\big) - [z_2 \cdot (x_2 + \mu^B e)]_j \Big\} \\ - \sum_{i=1}^{m_F} \Big\{ \mu^B [w^E 1]_i \ln\big([w_1]_i [s_1 + \mu^B e]_i^2\big) - [w_1 \cdot (s_1 + \mu^B e)]_i \Big\} \\ - \sum_{i=1}^{m_F} \Big\{ \mu^B [w^E 2]_i \ln\big([w_2]_i [s_2 + \mu^B e]_i^2\big) - [w_2 \cdot (s_2 + \mu^B e)]_i \Big\}. \qquad \text{(A.4.2)} \end{aligned}$$

The gradient $\nabla M(x,\, x_1,\, x_2,\, s,\, s_1,\, s_2,\, y,\, v,\, z_1,\, z_2,\, w_1,\, w_2)$ may be defined in terms of the quantities $X_1^\mu = \operatorname{diag}(E_L x - \ell^X + \mu^B e)$, $X_2^\mu = \operatorname{diag}(u^X - E_U x + \mu^B e)$, $Z_1 = \operatorname{diag}(z_1)$, $Z_2 = \operatorname{diag}(z_2)$, $W_1 = \operatorname{diag}(w_1)$, $W_2 = \operatorname{diag}(w_2)$, $S^\mu 1 = \operatorname{diag}(L_L s - \ell^S + \mu^B e)$ and

$S^\mu 2 = \operatorname{diag}(u^S - L_U s + \mu^B e)$, in particular

$$\nabla M = \begin{pmatrix} g - A^{\mathrm{T}}\Big(2(v^E + \frac{1}{\mu^A}(Ax-b)) - v\Big) - J^T\Big(2(y^E - \frac{1}{\mu^P}(c-s)) - y\Big) \\ z_1 - 2\mu^B (X_1^\mu)^{-1} z^E 1 \\ z_2 - 2\mu^B (X_2^\mu)^{-1} z^E 2 \\ 2\Big(y^E - \frac{1}{\mu^P}(c-s)\Big) - y \\ w_1 - 2\mu^B (S^\mu 1)^{-1} w^E 1 \\ w_2 - 2\mu^B (S^\mu 2)^{-1} w^E 2 \\ c - s + \mu^P (y - y^E) \\ Ax - b + \mu^A (v - v^E) \\ x_1 + \mu^B e - \mu^B Z_1^{-1} z^E 1 \\ x_2 + \mu^B e - \mu^B Z_2^{-1} z^E 2 \\ s_1 + \mu^B e - \mu^B W_1^{-1} w^E 1 \\ s_2 + \mu^B e - \mu^B W_2^{-1} w^E 2 \end{pmatrix}$$

$$= \begin{pmatrix} g - A^{\mathrm{T}}\Big(2(v^E + \frac{1}{\mu^A}(Ax-b)) - v\Big) - J^T\Big(2(y^E - \frac{1}{\mu^P}(c-s)) - y\Big) \\ (X_1^\mu)^{-1}\Big(z_1 \cdot x_1 + \mu^B z^E 1 + \mu^B (z_1 - z^E 1)\Big) \\ (X_2^\mu)^{-1}\Big(z_2 \cdot x_2 + \mu^B z^E 2 + \mu^B (z_2 - z^E 2)\Big) \\ 2\Big(y^E - \frac{1}{\mu^P}(c-s)\Big) - y \\ (S^\mu 1)^{-1}\Big(w_1 \cdot s_1 + \mu^B w^E 1 + \mu^B (w_1 - w^E 1)\Big) \\ (S^\mu 2)^{-1}\Big(w_2 \cdot s_2 + \mu^B w^E 2 + \mu^B (w_2 - w^E 2)\Big) \\ c - s + \mu^P (y - y^E) \\ Ax - b + \mu^A (v - v^E) \\ Z_1^{-1}\Big(z_1 \cdot x_1 + \mu^B (z_1 - z^E 1)\Big) \\ Z_2^{-1}\Big(z_2 \cdot x_2 + \mu^B (z_2 - z^E 2)\Big) \\ W_1^{-1}\Big(w_1 \cdot s_1 + \mu^B (w_1 - w^E 1)\Big) \\ W_2^{-1}\Big(w_2 \cdot s_2 + \mu^B (w_2 - w^E 2)\Big) \end{pmatrix} = \begin{pmatrix} g - A^{\mathrm{T}}\Big(\pi^V + (\pi^V - v)\Big) - J^{\mathrm{T}}\Big(\pi^Y + (\pi^Y - y)\Big) \\ -\Big(\pi_1^Z + (\pi_1^Z - z_1)\Big) \\ -\Big(\pi_2^Z + (\pi_2^Z - z_2)\Big) \\ \pi^Y + (\pi^Y - y) \\ -\Big(\pi_1^W + (\pi_1^W - w_1)\Big) \\ -\Big(\pi_2^W + (\pi_2^W - w_2)\Big) \\ -D_Y (\pi^Y - y) \\ -D_A (\pi^V - v) \\ -D_Z 1 (\pi_1^Z - z_1) \\ -D_Z 2 (\pi_2^Z - z_2) \\ -D_W 1 (\pi_1^W - w_1) \\ -D_W 2 (\pi_2^W - w_2) \end{pmatrix},$$

where the quantities D_Y, π^Y, D_A, π^V, $D_W 1$, $D_W 2$, π_1^W, π_2^W, $D_Z 1$, $D_Z 2$, π_1^Z, and π_2^Z are defined in (A.3.5).

The Hessian $\nabla^2 M(x, x_1, x_2, s, s_1, s_2, y, v, z_1, z_2, w_1, w_2)$ is given by

$$\begin{pmatrix}
H_1 & 0 & 0 & -2J^{\mathrm{T}}D_Y^{-1} & 0 & 0 & J^{\mathrm{T}} & A^T & 0 & 0 & 0 & 0 \\
0 & 2\mathit{\Pi}^Z 1(X_1^\mu)^{-1} & 0 & 0 & 0 & 0 & -I_m & 0 & I_L^x & 0 & 0 & 0 \\
0 & 0 & 2\mathit{\Pi}^Z 2(X_2^\mu)^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & I_U^x & 0 & 0 \\
-2D_Y^{-1}J & 0 & 0 & 2D_Y^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2\mathit{\Pi}^W 1(S^\mu 1)^{-1} & 0 & 0 & 0 & 0 & 0 & I_L^s & 0 \\
0 & 0 & 0 & 0 & 0 & 2\mathit{\Pi}^W 2(S^\mu 2)^{-1} & 0 & 0 & 0 & 0 & 0 & I_U^s \\
J & 0 & 0 & -I_m & 0 & 0 & D_Y & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\
0 & I_L^x & 0 & 0 & 0 & 0 & 0 & 0 & X_1^\mu Z_1^{-2}\mathit{\Pi}^Z 1 & 0 & 0 & 0 \\
0 & 0 & I_U^x & 0 & 0 & 0 & 0 & 0 & 0 & X_1^\mu Z_2^{-2}\mathit{\Pi}^Z 2 & 0 & 0 \\
0 & 0 & 0 & 0 & I_L^s & 0 & 0 & 0 & 0 & 0 & S^\mu 1 W_1^{-2}\mathit{\Pi}^W 1 & 0 \\
0 & 0 & 0 & 0 & 0 & I_U^s & 0 & 0 & 0 & 0 & 0 & S^\mu 2 W_2^{-2}\mathit{\Pi}^W 2
\end{pmatrix},$$

where

$$H_1 = H(x, 2\pi^Y - y) + \frac{2}{\mu^A}A^{\mathrm{T}}A + \frac{2}{\mu^P}J^{\mathrm{T}}J = H(x, 2\pi^Y - y) + 2A^{\mathrm{T}}D_A^{-1}A + 2J^{\mathrm{T}}D_Y^{-1}J,$$

and I_L^x, I_L^x, I_L^s, and I_U^s denote identity matrices of dimension n_L, n_U, m_L and m_U respectively. The usual convention regarding diagonal matrices formed from vectors applies, with $\mathit{\Pi}^Z 1 = \mathrm{diag}(\pi_1^Z)$, $\mathit{\Pi}^Z 2 = \mathrm{diag}(\pi_2^Z)$, $\mathit{\Pi}^W 1 = \mathrm{diag}(\pi_1^W)$, and $\mathit{\Pi}^W 2 = \mathrm{diag}(\pi_2^W)$.

A.5 Derivation of the shifted primal-dual penalty-barrier direction

The primal-dual penalty-barrier problem (A.4.1) may be written in the form

$$\underset{p\in\mathcal{I}}{\text{minimize}}\;\; M(p) \quad \text{subject to} \quad Cp = b_C,$$

where

$$\mathcal{I} = \big\{p : p = (x, x_1, x_2, s, s_1, s_2, y, v, z_1, z_2, w_1, w_2), \;\; \text{with } x_i + \mu^B e > 0,\; s_i + \mu^B e > 0,\; z_i > 0,\; w_i > 0 \text{ for } i = 1, 2\big\},$$

and

$$C = \begin{pmatrix} E_X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_L & -I^x_L & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_U & 0 & I^x_U & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_L & -I^s_L & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_U & 0 & I^s_U & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \text{with} \quad b_C = \begin{pmatrix} b_X \\ \ell^X \\ u^X \\ h_X \\ \ell^S \\ u^S \end{pmatrix}. \tag{A.5.1}$$

Let $p \in \mathcal{I}$ be given such that $Cp = b_C$. The Newton direction Δv is given by the solution of the subproblem

$$\underset{\Delta v}{\text{minimize}} \;\; \nabla M(p)^{\mathrm{T}} \Delta v + \tfrac{1}{2} \Delta v^{\mathrm{T}} \nabla^2 M(p) \Delta v \quad \text{subject to} \;\; C \Delta v = b_C - Cp = 0. \tag{A.5.2}$$

However, instead of solving (A.5.2), we formulate a linearly constrained *approximate* Newton method by approximating the Hessian $\nabla^2 M$ by a positive-definite matrix B. Consider the matrix defined by replacing π^Y by y, π^Z_1 by z_1, π^Z_2 by z_2, π^W_1 by w_1, π^W_2 by w_2 in the matrix $\nabla^2 M(x,\, x_1,\, x_2,\, s,\, s_1,\, s_2,\, y,\, v,\, z_1,\, z_2,\, w_1,\, w_2)$. This gives an approximate Hessian $B(x,\, x_1,\, x_2,\, s,\, s_1,\, s_2,\, y,\, v,\, z_1,\, z_2,\, w_1,$

w_2) of the form

$$\begin{pmatrix}
H^B + 2A^{\mathrm{T}}D_A^{-1}A + 2J^{\mathrm{T}}D_Y^{-1}J & 0 & 0 & -2J^{\mathrm{T}}D_Y^{-1} & 0 & 0 & J^{\mathrm{T}} & A^{\mathrm{T}} & 0 & 0 & 0 & 0 \\
0 & 2(D_Z1)^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & I_L^x & 0 & 0 & 0 \\
0 & 0 & 2(D_Z2)^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & I_U^x & 0 & 0 \\
-2D_Y^{-1}J & 0 & 0 & 2D_Y^{-1} & 0 & 0 & -I_m & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2(D_W1)^{-1} & 0 & 0 & 0 & 0 & 0 & I_L^s & 0 \\
0 & 0 & 0 & 0 & 0 & 2(D_W2)^{-1} & 0 & 0 & 0 & 0 & 0 & I_U^s \\
J & 0 & 0 & -I_m & 0 & 0 & D_Y & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\
0 & I_L^x & 0 & 0 & 0 & 0 & 0 & 0 & D_Z1 & 0 & 0 & 0 \\
0 & 0 & I_U^x & 0 & 0 & 0 & 0 & 0 & 0 & D_Z2 & 0 & 0 \\
0 & 0 & 0 & 0 & I_L^s & 0 & 0 & 0 & 0 & 0 & D_W1 & 0 \\
0 & 0 & 0 & 0 & 0 & I_U^s & 0 & 0 & 0 & 0 & 0 & D_W2
\end{pmatrix},$$

where $H^B \approx H(x,y)$ is chosen so that the approximate Hessian B is positive definite. Given $B(p)$, with $p = (x,\, x_1,\, x_2,\, s,\, s_1,\, s_2,\, y,\, v,$ $z_1,\, z_2,\, w_1,\, w_2)$, an approximate Newton direction is given by the solution of the QP subproblem

$$\underset{\Delta v}{\text{minimize}} \;\; \nabla M(p)^T \Delta v + \tfrac{1}{2}\Delta v^{\mathrm{T}} B(p) \Delta v \quad \text{subject to} \;\; C\Delta v = 0.$$

Let N denote a matrix whose columns form a basis for null(C), i.e., the columns of N are linearly independent and $CN = 0$. Every feasible direction Δv may be written in the form $\Delta v = Nd$. This implies that d satisfies the reduced equations $N^{\mathrm{T}}B(p)Nd = -N^T\nabla M(p)$.

Consider the null-space basis defined from the columns of the matrix

$$N = \begin{pmatrix} E_F^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_F^T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_m & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_A & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_L^x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_U^x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_L^s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_U^s \end{pmatrix}, \tag{A.5.3}$$

where $E_{LF} = E_L E_F^T$, $E_{UF} = E_U E_F^T$, $L_{LF} = L_L L_F^T$ and $L_{UF} = L_U L_F^T$. The definition of N of (A.5.3) gives the reduced approximate Hessian $N^{\mathrm{T}} B(p) N$ such that

$$\begin{pmatrix} \widehat{H}_F & -2J_F^T D_Y^{-1} L_F^T & J_F^T & A_F^T & E_{LF}^T & -E_{UF}^T & 0 & 0 \\ -2L_F D_Y^{-1} J_F & 2L_F \big(D_Y^{-1} + D_W^{\dagger}\big) L_F^T & -L_F & 0 & 0 & 0 & L_{LF}^T & L_{UF}^T \\ J_F & -L_F^T & D_Y & 0 & 0 & 0 & 0 & 0 \\ A_F & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & D_Z1 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & D_Z2 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & D_W1 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & D_W2 \end{pmatrix},$$

where $J_F = J E_F^T$, $A_F = A E_F^T$, $\widehat{H}_F = E_F H^B E_F^T + 2A_F^{\mathrm{T}} D_A^{-1} A_F + 2J_F^{\mathrm{T}} D_Y^{-1} J_F + 2E_F D_Z^{\dagger} E_F^T$, with

$$D_Z^{\dagger} = E_L^T (D_Z1)^{-1} E_L + E_U^T (D_Z2)^{-1} E_U, \quad \text{and} \quad D_W^{\dagger} = L_L^T \big(D_W1\big)^{-1} L_L + L_U^T \big(D_W2\big)^{-1} L_U,$$

Similarly, the reduced gradient $N^T \nabla M(p)$ is given by

$$\begin{pmatrix} g_F - A_F^{\mathrm{T}}\big(2\pi^V - v\big) - J_F^{\mathrm{T}}\big(2\pi^Y - y\big) - E_{LF}^T(2\pi_1^Z - z_1) + E_{UF}^T(2\pi_2^Z - z_2) \\ 2\pi_F^Y - y_F - L_{LF}^T(2\pi_1^W - w_1) + L_{UF}^T(2\pi_2^W - w_2) \\ -D_Y(\pi^Y - y) \\ -D_A(\pi^V - v) \\ -D_Z 1(\pi_1^Z - z_1) \\ -D_Z 2(\pi_2^Z - z_2) \\ -D_W 1(\pi_1^W - w_1) \\ -D_W 2(\pi_2^W - w_2) \end{pmatrix},$$

where $g_F = E_F g$, $\pi_F^Y = L_F \pi^Y$ and $y_F = L_F y$. The reduced approximate Newton equations $N^{\mathrm{T}} B(p) N d = -N^T \nabla M(p)$ are then

$$\begin{pmatrix} \widehat{H}_F & -2J_F^T D_Y^{-1} L_F^T & J_F^T & A_F^T & E_{LF}^T & -E_{UF}^T & 0 & 0 \\ -2L_F D_Y^{-1} J_F & 2L_F(D_Y^{-1} + D_W^{\dagger}) L_F^T & -L_F & 0 & 0 & 0 & L_{LF}^T & L_{UF}^T \\ J_F & -L_F^T & D_Y & 0 & 0 & 0 & 0 & 0 \\ A_F & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & D_Z 1 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & D_Z 2 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & D_W 1 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & D_W 2 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{pmatrix}$$

$$= - \begin{pmatrix} g_F - A_F^{\mathrm{T}}\big(2\pi^V - v\big) - J_F^{\mathrm{T}}\big(2\pi^Y - y\big) - E_{LF}^T(2\pi_1^Z - z_1) + E_{UF}^T(2\pi_2^Z - z_2) \\ 2\pi_F^Y - y_F - L_{LF}^T(2\pi_1^W - w_1) + L_{UF}^T(2\pi_2^W - w_2) \\ -D_Y(\pi^Y - y) \\ -D_A(\pi^V - v) \\ -D_Z 1(\pi_1^Z - z_1) \\ -D_Z 2(\pi_2^Z - z_2) \\ -D_W 1(\pi_1^W - w_1) \\ -D_W 2(\pi_2^W - w_2) \end{pmatrix}, \quad \text{(A.5.4)}$$

Given any nonsingular matrix R, the direction d satisfies $RN^{\mathrm{T}}B(p)Nd = -RN^{T}\nabla M(p)$. In particular, if R is the block upper-triangular matrix R such that

$$R = \begin{pmatrix} I_F^x & 0 & -2J_F^TD_Y^{-1} & -2A_F^TD_A^{-1} & -2E_{LF}^T(D_Z1)^{-1} & 2E_{UF}^T(D_Z2)^{-1} & 0 & 0 \\ & I_F^s & 2L_FD_Y^{-1} & 0 & 0 & 0 & -2L_{LF}^T(D_W1)^{-1} & 2L_{UF}^T(D_W2)^{-1} \\ & & I_m & 0 & 0 & 0 & 0 & 0 \\ & & & I_A & 0 & 0 & 0 & 0 \\ & & & & I_{LF}^x & 0 & 0 & 0 \\ & & & & & I_{UF}^x & 0 & 0 \\ & & & & & & I_{LF}^s & 0 \\ & & & & & & & I_{UF}^s \end{pmatrix},$$

then

$$RN^{\mathrm{T}}B(p)N = \begin{pmatrix} H_F^B & 0 & -J_F^T & -A_F^T & -E_{LF}^T & E_{UF}^T & 0 & 0 \\ 0 & 0 & L_F & 0 & 0 & 0 & -L_{LF}^T & L_{UF}^T \\ J_F & -L_F^T & D_Y & 0 & 0 & 0 & 0 & 0 \\ A_F & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & D_Z1 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & D_Z2 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & D_W1 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & D_W2 \end{pmatrix},$$

and

$$RN^T\nabla M(p) = \begin{pmatrix} g_F - J_F^{\mathrm{T}}y - A_F^T v - z_F \\ y_F - w_F \\ -D_Y(\pi^Y - y) \\ -D_A(\pi^V - v) \\ -D_Z1(\pi_1^Z - z_1) \\ -D_Z2(\pi_2^Z - z_2) \\ -D_W1(\pi_1^W - w_1) \\ -D_W2(\pi_2^W - w_2) \end{pmatrix}.$$

The matrix R is nonsingular because Z_1, Z_2 and W are positive definite, which implies that we may solve the following (unsymmetric) reduced approximate Newton equations for d:

$$\begin{pmatrix} H_F^B & 0 & -J_F^T & -A_F^T & -E_{LF}^T & E_{UF}^T & 0 & 0 \\ 0 & 0 & L_F & 0 & 0 & 0 & -L_{LF}^T & L_{UF}^T \\ J_F & -L_F^T & D_Y & 0 & 0 & 0 & 0 & 0 \\ A_F & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & D_Z1 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & D_Z2 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & D_W1 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & D_W2 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{pmatrix} = - \begin{pmatrix} g_F - J_F^{\mathrm{T}} y - A_F^T v - z_F \\ y_F - w_F \\ -D_Y(\pi^Y - y) \\ -D_A(\pi^V - v) \\ -D_Z1(\pi_1^Z - z_1) \\ -D_Z2(\pi_2^Z - z_2) \\ -D_W1(\pi_1^W - w_1) \\ -D_W2(\pi_2^W - w_2) \end{pmatrix}. \qquad (A.5.5)$$

Then, the expression $\varDelta v = Nd$ implies that

$$\varDelta v = \begin{pmatrix} \varDelta x \\ \varDelta x_1 \\ \varDelta x_2 \\ \varDelta s \\ \varDelta s_1 \\ \varDelta s_2 \\ \varDelta y \\ \varDelta v \\ \varDelta z_1 \\ \varDelta z_2 \\ \varDelta w_1 \\ \varDelta w_2 \end{pmatrix} = Nd = \begin{pmatrix} E_F^T d_1 \\ d_1 \\ -d_1 \\ L_F^T d_2 \\ d_2 \\ -d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{pmatrix}. \tag{A.5.6}$$

These identities allow us to write equations (A.5.5) in the form

$$\begin{pmatrix} H_F^B & 0 & -J_F^T & -A_F^T & -E_{LF}^T & E_{UF}^T & 0 & 0 \\ 0 & 0 & L_F & 0 & 0 & 0 & -L_{LF}^T & L_{UF}^T \\ J_F & -L_F^T & D_Y & 0 & 0 & 0 & 0 & 0 \\ A_F & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & D_Z1 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & D_Z2 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & D_W1 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & D_W2 \end{pmatrix} \begin{pmatrix} \Delta x_F \\ \Delta s_F \\ \Delta y \\ \Delta v \\ \Delta z_1 \\ \Delta z_2 \\ \Delta w_1 \\ \Delta w_2 \end{pmatrix} = - \begin{pmatrix} g_F - J_F^{\mathrm{T}} y - A_F^T v - z_F \\ y_F - w_F \\ -D_Y(\pi^Y - y) \\ -D_A(\pi^V - v) \\ -D_Z1(\pi_1^Z - z_1) \\ -D_Z2(\pi_2^Z - z_2) \\ -D_W1(\pi_1^W - w_1) \\ -D_W2(\pi_2^W - w_2) \end{pmatrix}, \tag{A.5.7}$$

with $\Delta x = E_F^T \Delta x_F$, $\Delta s = L_F^T \Delta s_F$, $\Delta x_1 = \Delta x_F - (\ell^X - E_L x + x_1)$, $\Delta x_2 = -\Delta x_F + (u^X - E_U x - x_2)$, $\Delta s_1 = \Delta s_F - (\ell^S - L_L s + s_1)$ and $\Delta s_2 = -\Delta s_F + (u^S - L_U s - s_2)$,

The shifted penalty-barrier equations (A.5.7) are the same as the path following equations (A.3.4) except for the $(1, 1)$ block, where $H(x, y)$ replaced by $H^B(x, y)$.

A.6 The primal-dual cubic regularization direction

Given a vector of primal-dual variables $p = (x,\ x_1,\ x_2,\ s,\ s_1,\ s_2,\ y,\ v,\ z_1,\ z_2,\ w_1,\ w_2)$, each iteration of a cubic regularization method for solving (A.1.1) involves finding a vector Δv of the form $\Delta v = Nd$, where N is a basis for the null-space of the matrix C of

(A.5.1), and d is an approximate solution of the subproblem

$$\underset{d}{\text{minimize}}\ \ \mathcal{C}(d) = g_N^T d + \tfrac{1}{2} d^{\mathrm{T}} B_N(p) d + \frac{1}{3}\mu^C \|d\|_T^3 \tag{A.6.1}$$

where g_N and B_N are the reduced gradient and reduced Hessian $g_N = N^T \nabla M$ and $B_N(p) = N^{\mathrm{T}} B(p) N$, $\|d\|_T = (d^T T d)^{1/2}$, $\mu^C \geq 0$ is the cubic regularization parameter, and T is a positive-definite diagonal matrix. The subproblem (A.6.1) may be written as

$$\underset{\Delta v_M}{\text{minimize}}\ \ g_N^T T^{-1/2} \Delta v_M + \tfrac{1}{2} \Delta v_M^T T^{-1/2} B_N(p) T^{-1/2} \Delta v_M + \frac{1}{3}\mu^C \|\Delta v_M\|_2, \tag{A.6.2}$$

where $\Delta v_M = T^{1/2} d$. The application of the method of Moré and Sorensen [34] to solve the subproblem (A.6.2) requires the solution of the so-called *secular equations*, which have the form

$$(\bar{B}_N + \sigma I)\Delta v_M = -\bar{g}_N, \tag{A.6.3}$$

with σ a nonnegative scalar, $\bar{B}_N = T^{-1/2} B_N(p) T^{-1/2}$, and $\bar{g}_N = T^{-1/2} g_N$. In this note we consider the solution of the related equations

$$(B_N + \sigma T) d = -g_N, \tag{A.6.4}$$

and recover the solution of the secular equations (A.6.3) from the computed vector d.

The identity (A.5.6) allows the solution of the approximate Newton equations $B_N(p) d = -g_N$ (A.5.4) to be written in terms of the change in the variables $(x,\, s,\, s_1,\, s_2,\, y,\, v,\, z_1,\, z_2,\, w_1,\, w_2)$. In particular, we have

$$\begin{pmatrix} \widehat{H}_F & -2J_F^T D_Y^{-1} L_F^T & J_F^T & A_F^T & E_{LF}^T & -E_{UF}^T & 0 & 0 \\ -2L_F D_Y^{-1} J_F & 2L_F(D_Y^{-1} + D_W^{\dagger})L_F^T & -L_F & 0 & 0 & 0 & L_{LF}^T & L_{UF}^T \\ J_F & -L_F^T & D_Y & 0 & 0 & 0 & 0 & 0 \\ A_F & 0 & 0 & D_A & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & D_Z 1 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & D_Z 2 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & D_W 1 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & D_W 2 \end{pmatrix} \begin{pmatrix} \Delta x_F \\ \Delta s_F \\ \Delta y \\ \Delta v \\ \Delta z_1 \\ \Delta z_2 \\ \Delta w_1 \\ \Delta w_2 \end{pmatrix}$$

$$= - \begin{pmatrix} g_F - A_F^{\mathrm{T}}\big(2\pi^V - v\big) - J_F^{\mathrm{T}}\big(2\pi^Y - y\big) - E_{LF}^T(2\pi_1^Z - z_1) + E_{UF}^T(2\pi_2^Z - z_2) \\ 2\pi_F^Y - y_F - L_{LF}^T(2\pi_1^W - w_1) + L_{UF}^T(2\pi_2^W - w_2) \\ -D_Y(\pi^Y - y) \\ -D_A(\pi^V - v) \\ -D_Z 1(\pi_1^Z - z_1) \\ -D_Z 2(\pi_2^Z - z_2) \\ -D_W 1(\pi_1^W - w_1) \\ -D_W 2(\pi_2^W - w_2) \end{pmatrix},$$

where

$$\widehat{H}_F = H_F + J_F^T D_Y^{-1} J_F + A_F^T D_A^{-1} A_F + E_F D_Z^{\dagger} E_F^T,$$

with

$$
\begin{array}{llll}
D_Y = \mu^P I_m, & \pi^Y = y^E - \dfrac{1}{\mu^P}(c - s), & D_A = \mu^A I_A, & \pi^V = v^E - \dfrac{1}{\mu^A}(Ax - b), \\
D_W 1 = S^\mu 1 W_1^{-1}, & \pi_1^W = \mu^B (S^\mu 1)^{-1} w^E 1, & D_Z 1 = X_1^\mu Z_1^{-1}, & \pi_1^Z = \mu^B (X_1^\mu)^{-1} z^E 1, \\
D_W 2 = S^\mu 2 W_2^{-1}, & \pi_2^W = \mu^B (S^\mu 2)^{-1} w^E 2, & D_Z 2 = X_2^\mu Z_2^{-1}, & \pi_2^Z = \mu^B (X_2^\mu)^{-1} z^E 2, \\
 & \pi^W = L_L^T \pi_1^W - L_U^T \pi_2^W & & \pi^Z = E_L^T \pi_1^Z - E_U^T \pi_2^Z.
\end{array}
$$

In the cubic regularization case, we make no assumption that B is positive definite, i.e., $H_F = E_F H(x,y) E_F^T$ with $H(x,y)$ the Hessian of the Lagrangian function.

The first step in the formulation of the cubic regularization equations (A.6.4) and their solution is to write the reduced gradient and Hessian of the merit function in terms of the vectors $\vec{x}$ and $\vec{y}$ that combine the primal variables (x, s) and dual variables $(y, v, z_1, z_2, w_1, w_2)$. Let $\vec{g}$, $\vec{H}$, $\vec{J}$ and $\vec{D}$ denote the quantities

$$
\vec{g} = \begin{pmatrix} g_F \\ 0 \end{pmatrix}, \quad \vec{H} = \begin{pmatrix} H_F & 0 \\ 0 & 0 \end{pmatrix}, \quad \vec{J} = \begin{pmatrix} J_F & -L_F^T \\ A_F & 0 \\ E_{LF} & 0 \\ -E_{UF} & 0 \\ 0 & L_{LF} \\ 0 & -L_{UF} \end{pmatrix} \quad \text{and} \quad \vec{D} = \begin{pmatrix} D_Y & 0 & 0 & 0 & 0 & 0 \\ 0 & D_A & 0 & 0 & 0 & 0 \\ 0 & 0 & D_Z 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & D_Z 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & D_W 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & D_W 2 \end{pmatrix}.
$$

Similarly, let $\vec{T}_x = \operatorname{diag}(T^x, T^s)$ and $\vec{T}_y = \operatorname{diag}(T^y,\, T^v,\, T^z 1,\, T^z 2,\, T^w 1,\, T^w 2)$. The equations $(B_N + \sigma T)\Delta v = -g_N$ may be written in the form

$$\begin{pmatrix} \vec{H} + 2\vec{J}^T\vec{D}^{-1}\vec{J} + \sigma\vec{T}_x & \vec{J}^T \\ \vec{J} & \vec{D} + \sigma\vec{T}_y \end{pmatrix} \begin{pmatrix} \Delta\vec{x} \\ \Delta\vec{y} \end{pmatrix} = - \begin{pmatrix} \vec{g} - \vec{J}^T\vec{\pi} - \vec{J}^T(\vec{\pi} - \vec{y}) \\ -\vec{D}(\vec{\pi} - \vec{y}) \end{pmatrix}, \tag{A.6.5}$$

where

$$\vec{y} = \begin{pmatrix} y \\ v \\ z_1 \\ z_2 \\ w_1 \\ w_2 \end{pmatrix}, \quad \vec{\pi} = \begin{pmatrix} \pi^Y \\ \pi^V \\ \pi_1^Z \\ \pi_2^Z \\ \pi_1^W \\ \pi_2^W \end{pmatrix}, \quad \Delta\vec{x} = \begin{pmatrix} \Delta x \\ \Delta s \end{pmatrix}, \quad \text{and} \quad \Delta\vec{y} = \begin{pmatrix} \Delta y \\ \Delta z \\ \Delta w \end{pmatrix}.$$

Applying the nonsingular matrix $\begin{pmatrix} I & -2\vec{J}^T\vec{D}^{-1} \\ & I \end{pmatrix}$ to both sides of (A.6.5) gives the equivalent system

$$\begin{pmatrix} \vec{H} + \sigma\vec{T}_x & -\vec{J}^T(I + 2\sigma\vec{D}^{-1}\vec{T}_y) \\ \vec{J} & \vec{D} + \sigma\vec{T}_y \end{pmatrix} \begin{pmatrix} \Delta\vec{x} \\ \Delta\vec{y} \end{pmatrix} = - \begin{pmatrix} \vec{g} - \vec{J}^{\mathrm{T}}\vec{y} \\ \vec{D}(\vec{y} - \vec{\pi}) \end{pmatrix}.$$

As in Gertz and Gill [21], we set $\vec{T}_x = I$ and $\vec{T}_y = \vec{D}$. With this choice, the associated vectors $\Delta\vec{x}$ and $\Delta\vec{y}$ satisfy the equations

$$\begin{pmatrix} \vec{H} + \sigma I & -\vec{J}^T \\ \vec{J} & \bar{\sigma}\vec{D} \end{pmatrix} \begin{pmatrix} \Delta\vec{x} \\ (1 + 2\sigma)\Delta\vec{y} \end{pmatrix} = - \begin{pmatrix} \vec{g} - \vec{J}^{\mathrm{T}}\vec{y} \\ \vec{D}(\vec{y} - \vec{\pi}) \end{pmatrix}, \tag{A.6.6}$$

where $\bar{\sigma} = (1+\sigma)/(1+2\sigma)$. In terms of the original variables, the unsymmetric equations (A.6.6) are

$$\begin{pmatrix} H_F + \sigma I_F^x & 0 & -J_F^T & -A_F^T & -E_{LF}^T & E_{UF}^T & 0 & 0 \\ 0 & \sigma I_F^s & L_F & 0 & 0 & 0 & -L_{LF}^T & L_{UF}^T \\ J_F & -L_F^T & \bar{\sigma} D_Y & 0 & 0 & 0 & 0 & 0 \\ A_F & 0 & 0 & \bar{\sigma} D_A & 0 & 0 & 0 & 0 \\ E_{LF} & 0 & 0 & 0 & \bar{\sigma} D_Z 1 & 0 & 0 & 0 \\ -E_{UF} & 0 & 0 & 0 & 0 & \bar{\sigma} D_Z 2 & 0 & 0 \\ 0 & L_{LF} & 0 & 0 & 0 & 0 & \bar{\sigma} D_W 1 & 0 \\ 0 & -L_{UF} & 0 & 0 & 0 & 0 & 0 & \bar{\sigma} D_W 2 \end{pmatrix} \begin{pmatrix} \Delta x_F \\ \Delta s_F \\ (1+2\sigma)\Delta y \\ (1+2\sigma)\Delta v \\ (1+2\sigma)\Delta z_1 \\ (1+2\sigma)\Delta z_2 \\ (1+2\sigma)\Delta w_1 \\ (1+2\sigma)\Delta w_2 \end{pmatrix}$$

$$= - \begin{pmatrix} E_F\big(g - J^{\mathrm{T}} y - A^{\mathrm{T}} v - z\big) \\ L_F\big(y - w\big) \\ c(x) - s + \mu^P (y - y^E) \\ Ax - b + \mu^A (v - v^E) \\ Z_1^{-1}\big(z_1 \cdot (E_L x - \ell^X) + \mu^B (z_1 - z^E 1)\big) \\ Z_2^{-1}\big(z_2 \cdot (u^X - E_U x) + \mu^B (z_2 - z^E 2)\big) \\ W_1^{-1}\big(w_1 \cdot (L_L s - \ell^S) + \mu^B (w_1 - w^E 1)\big) \\ W_2^{-1}\big(w_2 \cdot (u^S - L_U s) + \mu^B (w_2 - w^E 2)\big) \end{pmatrix}, \quad \text{(A.6.7)}$$

where $\bar{\sigma} = (1+\sigma)/(1+2\sigma)$. Collecting terms and reordering the equations and unknowns, we obtain

$$\begin{pmatrix} \bar{\sigma}D_A & 0 & 0 & 0 & 0 & 0 & A_F & 0 \\ 0 & \bar{\sigma}D_Z1 & 0 & 0 & 0 & 0 & E_{LF} & 0 \\ 0 & 0 & \bar{\sigma}D_Z2 & 0 & 0 & 0 & -E_{UF} & 0 \\ 0 & 0 & 0 & \bar{\sigma}D_W1 & 0 & L_{LF} & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{\sigma}D_W2 & -L_{UF} & 0 & 0 \\ 0 & 0 & 0 & -L_{LF}^T & L_{UF}^T & \sigma I_F^s & 0 & L_F \\ -A_F^T & -E_{LF}^T & E_{UF}^T & 0 & 0 & 0 & H_F + \sigma I_F^x & -J_F^T \\ 0 & 0 & 0 & 0 & 0 & -L_F^T & J_F & \bar{\sigma}D_Y \end{pmatrix} \begin{pmatrix} \Delta\widetilde{v} \\ \Delta\widetilde{z}_1 \\ \Delta\widetilde{z}_2 \\ \Delta\widetilde{w}_1 \\ \Delta\widetilde{w}_2 \\ \Delta s_F \\ \Delta x_F \\ \Delta\widetilde{y} \end{pmatrix} = - \begin{pmatrix} D_A(v - \pi^V) \\ D_Z1(z_1 - \pi_1^Z) \\ D_Z2(z_2 - \pi_2^Z) \\ D_W1(w_1 - \pi_1^W) \\ D_W2(w_2 - \pi_2^W) \\ L_F\big(y - w\big) \\ E_F\big(g - J^{\mathrm{T}}y - A^{\mathrm{T}}v - z\big) \\ D_Y(y - \pi^Y) \end{pmatrix}, \qquad \text{(A.6.8)}$$

where $\bar{D}_A = \bar{\sigma}D_A$, $\bar{D}_W1 = \bar{\sigma}D_W1$, $\bar{D}_W2 = \bar{\sigma}D_W2$, $\bar{D}_Z1 = \bar{\sigma}D_Z1$, $\bar{D}_Z2 = \bar{\sigma}D_Z2$, $\bar{D}_Y = \bar{\sigma}D_Y$, $\Delta\widetilde{y} = (1+2\sigma)\Delta y$, $\Delta\widetilde{v} = (1+2\sigma)\Delta v$, $\Delta\widetilde{z}_1 = (1+2\sigma)\Delta z_1$, $\Delta\widetilde{z}_2 = (1+2\sigma)\Delta z_2$, $\Delta\widetilde{w}_1 = (1+2\sigma)\Delta w_1$, and $\Delta\widetilde{w}_2 = (1+2\sigma)\Delta w_2$. We define

$$\bar{D}_W = \big(L_L^T(\bar{D}_W1)^{-1}L_L + L_U^T(\bar{D}_W2)^{-1}L_U\big)^\dagger = \bar{\sigma}\big(L_L^T(D_W1)^{-1}L_L + L_U^T(D_W2)^{-1}L_U\big)^\dagger = \bar{\sigma}D_W,$$

with $D_W = \big(L_{LF}^T(D_W1)^{-1}L_{LF} + L_{UF}^T(D_W2)^{-1}L_{UF}\big)^\dagger$. Similarly, define

$$\breve{D}_W = \big(D_W^\dagger + \sigma\bar{\sigma}I_F^s\big)^\dagger.$$

Premultiplying the equations (A.6.8) by the matrix

$$\begin{pmatrix}
I_A & & & & & & & \\
0 & I^x_{LF} & & & & & & \\
0 & 0 & I^x_{UF} & & & & & \\
0 & 0 & 0 & I^s_{LF} & & & & \\
0 & 0 & 0 & 0 & I^s_{UF} & & & \\
0 & 0 & 0 & \frac{1}{\bar\sigma}L^T_{LF}(D_W1)^{-1} & -\frac{1}{\bar\sigma}L^T_{UF}(D_W2)^{-1} & I^s_F & & \\
\frac{1}{\bar\sigma}A^T_F D_A^{-1} & \frac{1}{\bar\sigma}E^T_{LF}(D_Z1)^{-1} & -\frac{1}{\bar\sigma}E^T_{UF}(D_Z2)^{-1} & 0 & 0 & 0 & I^x_F & \\
0 & 0 & 0 & \breve{D}_W L^T_L(D_W1)^{-1} & -\breve{D}_W L^T_U(D_W2)^{-1} & \bar\sigma\breve{D}_W L^T_F & 0 & I_m
\end{pmatrix}$$

gives the block upper-triangular system

$$\begin{pmatrix} \bar\sigma D_A & 0 & 0 & 0 & 0 & 0 & A_F & 0 \\ 0 & \bar\sigma D_Z 1 & 0 & 0 & 0 & 0 & E_{LF} & 0 \\ 0 & 0 & \bar\sigma D_Z 2 & 0 & 0 & 0 & -E_{UF} & 0 \\ 0 & 0 & 0 & \bar\sigma D_W 1 & 0 & L_{LF} & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar\sigma D_W 2 & -L_{UF} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\bar\sigma} L_F \breve{D}_W^{\dagger} L_F^T & 0 & L_F \\ 0 & 0 & 0 & 0 & 0 & 0 & \widehat{H}_F + \sigma I_F^x & -J_F^T \\ 0 & 0 & 0 & 0 & 0 & 0 & J_F & \bar\sigma(D_Y + \breve{D}_W) \end{pmatrix} \begin{pmatrix} \Delta\widetilde{v} \\ \Delta\widetilde{z}_1 \\ \Delta\widetilde{z}_2 \\ \Delta\widetilde{w}_1 \\ \Delta\widetilde{w}_2 \\ \Delta s_F \\ \Delta x_F \\ \Delta\widetilde{y} \end{pmatrix}$$

$$= - \begin{pmatrix} D_A(v - \pi^V) \\ D_Z 1(z_1 - \pi_1^Z) \\ D_Z 2(z_2 - \pi_2^Z) \\ D_W 1(w_1 - \pi_1^W) \\ D_W 2(w_2 - \pi_2^W) \\ L_F\Big(y - w + \dfrac{1}{\bar\sigma}\big[w - \pi^W\big]\Big) \\ E_F\Big(g - J^{\mathrm{T}} y - A^{\mathrm{T}} v - z + \dfrac{1}{\bar\sigma}\big[A^T(v - \pi^V) + z - \pi^Z\big]\Big) \\ D_Y(y - \pi^Y) + \breve{D}_W\big(\bar\sigma(y - w) + w - \pi^W\big) \end{pmatrix},$$

where

$$\widehat{H}_F = E_F\Big(H + \frac{1}{\bar{\sigma}}\big(J^{\mathrm{T}} D_Y^{-1} J + A^{\mathrm{T}} D_A^{-1} A + D_Z^{\dagger}\big)\Big) E_F^T,$$

$w = L_X^T w_X + L_L^T w_1 - L_U^T w_2$, $z = E_X^T z_X + E_L^T z_1 - E_U^T z_2$, $\pi^W = L_L^T \pi_1^W - L_U^T \pi_2^W$ and $\pi^Z = E_L^T \pi_1^Z - E_U^T \pi_2^Z$. Using block back-substitution, Δx_F and Δy may be computed by solving the equations

$$\begin{pmatrix} \widehat{H}_F + \sigma I_F^x & -J_F^T \\ J_F & \bar{\sigma}(D_Y + \breve{D}_W) \end{pmatrix} \begin{pmatrix} \Delta x_F \\ \Delta \widetilde{y} \end{pmatrix} = - \begin{pmatrix} E_F\left(g - J^{\mathrm{T}} y - A^{\mathrm{T}} v - z + \frac{1}{\bar{\sigma}}\left[A^T(v - \pi^V) + z - \pi^Z\right]\right) \\ D_Y\,(y - \pi^Y) + \breve{D}_W\,(\bar{\sigma}(y - w) + w - \pi^W) \end{pmatrix}.$$

The full vector Δx is then computed as $\Delta x = E_F^T \Delta x_F$. Using the identity $\Delta s = L_F^T \Delta s_F$ in the sixth block of equations gives

$$\Delta s = -\bar{\sigma}\breve{D}_W \left(y + (1 + 2\sigma)\Delta y - w + \frac{1}{\bar{\sigma}}\left[w - \pi^W\right]\right).$$

There are several ways of computing Δw_1 and Δw_2. Instead of using the block upper-triangular system above, we use the last two blocks of equations of (A.6.7) to give

$$\Delta w_1 = -\frac{1}{1+\sigma}(S^{\mu}1)^{-1}\big(w_1 \cdot (L_L(s + \Delta s) - \ell^S + \mu^B e) - \mu^B w^E 1\big) \text{ and}$$
$$\Delta w_2 = -\frac{1}{1+\sigma}(S^{\mu}2)^{-1}\big(w_2 \cdot (u^S - L_U(s + \Delta s) + \mu^B e) - \mu^B w^E 2\big).$$

Similarly, using (A.6.7) to solve for Δz_1 and Δz_2 yields

$$\Delta z_1 = -\frac{1}{1+\sigma}(X_1^\mu)^{-1}\big(z_1 \cdot (E_L(x+\Delta x) - \ell^X + \mu^B e) - \mu^B z^E 1\big) \text{ and}$$
$$\Delta z_2 = -\frac{1}{1+\sigma}(X_2^\mu)^{-1}\big(z_2 \cdot (u^X - E_U(x+\Delta x) + \mu^B e) - \mu^B z^E 2\big).$$

Similarly, using the first block of equations (A.6.8) to solve for Δv gives $\Delta v = -(v - \widehat{\pi}^V)/(1+\sigma)$, with $\widehat{\pi}^V = v^E - \frac{1}{\mu^A}\big(A(x+\Delta x) - b\big)$. Finally, the vectors Δw_X and Δz_X are recovered as $\Delta w_X = [y + \Delta y - w]_X$ and $\Delta z_X = [g + H\Delta x - J^{\mathrm{T}}(y+\Delta y) - z]_X$.

A.7 Summary: equations for the cubic regularization direction

The results of the preceding section implies that the solution of the path-following equations $F'(v_P)\Delta v_P = -F(v_P)$ with F and F' given by (A.3.2) and (A.3.3) may be computed as follows. Let x and s be given primal variables and slack variables such that $E_X x = b_X$, $L_X s = h_X$ with $\ell^X - \mu^B < E_L x$, $E_U x < u^X + \mu^B$, $\ell^S - \mu^B < L_L s$, $L_U s < u^S + \mu^B$. Similarly, let z_1, z_2, w_1, w_2 and y denotes dual variables such that $w_1 > 0$, $w_2 > 0$, $z_1 > 0$, and $z_2 > 0$. Consider the diagonal matrices $X_1^\mu = \mathrm{diag}(E_L x - \ell^X + \mu^B e)$, $X_2^\mu = \mathrm{diag}(u^X - E_U x + \mu^B e)$, $Z_1 = \mathrm{diag}(z_1)$, $Z_2 = \mathrm{diag}(z_2)$, $W_1 = \mathrm{diag}(w_1)$, $W_2 = \mathrm{diag}(w_2)$, $S^\mu 1 = \mathrm{diag}(L_L s - \ell^S + \mu^B e)$ and

$S^{\mu}2 = \operatorname{diag}(u^{S} - L_{U}s + \mu^{B}e)$. Given the quantities

$$\begin{aligned}
D_{Y} &= \mu^{P} I_{m}, & \pi^{Y} &= y^{E} - \frac{1}{\mu^{P}}(c - s),\\
D_{A} &= \mu^{A} I_{A}, & \pi^{V} &= v^{E} - \frac{1}{\mu^{A}}(Ax - b),\\
(D_{Z}1)^{-1} &= (X_{1}^{\mu})^{-1} Z_{1}, & (D_{W}1)^{-1} &= (S^{\mu}1)^{-1} W_{1},\\
(D_{Z}2)^{-1} &= (X_{2}^{\mu})^{-1} Z_{2}, & (D_{W}2)^{-1} &= (S^{\mu}2)^{-1} W_{2},\\
D_{Z} &= \big(E_{L}^{T}(D_{Z}1)^{-1}E_{L} + E_{U}^{T}(D_{Z}2)^{-1}E_{U}\big)^{\dagger}, & D_{W} &= \big(L_{L}^{T}(D_{W}1)^{-1}L_{L} + L_{U}^{T}(D_{W}2)^{-1}L_{U}\big)^{\dagger},\\
& & \breve{D}_{W} &= \big(D_{W}^{\dagger} + \sigma\bar{\sigma} I_{F}^{s}\big)^{\dagger},\\
\pi_{1}^{Z} &= \mu^{B}(X_{1}^{\mu})^{-1} z^{E}1, & \pi_{1}^{W} &= \mu^{B}(S^{\mu}1)^{-1} w^{E}1,\\
\pi_{2}^{Z} &= \mu^{B}(X_{2}^{\mu})^{-1} z^{E}2, & \pi_{2}^{W} &= \mu^{B}(S^{\mu}2)^{-1} w^{E}2,\\
\pi^{Z} &= E_{L}^{T}\pi_{1}^{Z} - E_{U}^{T}\pi_{2}^{Z}, & \pi^{W} &= L_{L}^{T}\pi_{1}^{W} - L_{U}^{T}\pi_{2}^{W},
\end{aligned}$$

solve the KKT system

$$\begin{pmatrix} H_{F}(x,y) + \sigma I_{F}^{x} + \frac{1}{\bar{\sigma}}A_{F}^{T}D_{A}^{-1}A_{F} + \frac{1}{\bar{\sigma}}E_{F}D_{Z}^{\dagger}E_{F}^{T} & -J_{F}(x)^{T} \\ J_{F}(x) & \bar{\sigma}\big(D_{Y} + \breve{D}_{W}\big) \end{pmatrix} \begin{pmatrix} \Delta x_{F} \\ \Delta\widetilde{y} \end{pmatrix}$$

$$= -\begin{pmatrix} E_{F}\big(g - J^{\mathrm{T}}y - A^{\mathrm{T}}v - z + \frac{1}{\bar{\sigma}}\big[A^{T}(v - \pi^{V}) + z - \pi^{Z}\big]\big) \\ D_{Y}\big(y - \pi^{Y}\big) + \breve{D}_{W}\big(\bar{\sigma}(y - w) + w - \pi^{W}\big) \end{pmatrix}.$$

Then

$$
\begin{array}{lll}
\Delta x = E_F^T \Delta x_F, & \widehat{x} = x + \Delta x, & \Delta z_1 = -\dfrac{1}{1+\sigma}(X_1^{\mu})^{-1}\big(z_1 \cdot (E_L\widehat{x} - \ell^X + \mu^B e) - \mu^B z^E 1\big), \\
 & & \Delta z_2 = -\dfrac{1}{1+\sigma}(X_2^{\mu})^{-1}\big(z_2 \cdot (u^X - E_U\widehat{x} + \mu^B e) - \mu^B z^E 2\big), \\
\Delta y = \Delta\widetilde{y}/(1+2\sigma), & \widehat{y} = y + \Delta y, & \Delta s \;= -\bar{\sigma}\breve{D}_W\Big(y + (1+2\sigma)\Delta y - w + \dfrac{1}{\bar{\sigma}}\big[w - \pi^W\big]\Big), \\
 & \widehat{s} = s + \Delta s, & \Delta w_1 = -\dfrac{1}{1+\sigma}(S^{\mu}1)^{-1}\big(w_1 \cdot (L_L\widehat{s} - \ell^S + \mu^B e) - \mu^B w^E 1\big), \\
 & & \Delta w_2 = -\dfrac{1}{1+\sigma}(S^{\mu}2)^{-1}\big(w_2 \cdot (u^S - L_U\widehat{s} + \mu^B e) - \mu^B w^E 2\big), \\
 & \widehat{\pi}^V = v^E - \dfrac{1}{\mu^A}(A\widehat{x} - b), & \Delta v \;= -\dfrac{1}{1+\sigma}\big(v - \widehat{\pi}^V\big), \\
 & w = L_X^T w_X + L_L^T w_1 - L_U^T w_2, & z = \; E_X^T z_X + E_L^T z_1 - E_U^T z_2, \\
 & \widehat{v} = v + \Delta v, & \Delta w_X = \; [\widehat{y} - w]_X, \\
 & & \Delta z_X = \; [g + H\Delta x - J^{\mathrm{T}}\widehat{y} - z]_X.
\end{array}
$$

A.8 Solution of the cubic regularization equations with an arbitrary right-hand-side

Moré and Sorensen define a routine $z_{\text{null}}(\cdot)$ that uses the Cholesky factors of $\bar{B}_N + \sigma I$ and the condition estimator proposed by Cline, Moler, Stewart and Wilkinson [11]. As the method of Gill, Kungurtsev and Robinson does not compute an explicit factorization of $\bar{B}_N + \sigma I$, we define $z_{\text{null}}(\cdot)$ using the condition estimator `DLACON` supplied with LAPACK [1]. This routine generates an approximate null vector using Higham's [32] modification of Hager's algorithm [31]. This routine uses matrix-vector products with $(B_N + \sigma T)^{-1}$, rather than a matrix factorization, to estimate $\|(\bar{B}_N + \sigma I)^{-1}\|_1$. By-products of the computation of $\|(\bar{B}_N + \sigma I)^{-1}\|_1$ are vectors v and w such that $w = (\bar{B}_N + \sigma I)^{-1} v$, $\|v\|_1 = 1$ and

$$\|(\bar{B}_N + \sigma I)^{-1} v\|_1 = \|w\|_1 \approx \|(\bar{B}_N + \sigma I)^{-1}\|_1 = \max_{\|u\|_1 = 1} \|(\bar{B}_N + \sigma I)^{-1} u\|_1.$$

Thus, unless $\|w\| = 0$, the vector $y = w/\|w\|$ is a unit approximate null vector from which we determine an appropriate z such that $\|\mathit{\Delta} v_M + z\|_T = \delta$.

The reduced cubic regularization equations with a general right-hand side are given by

$$\begin{pmatrix} \bar{\sigma} D_A & 0 & 0 & 0 & 0 & 0 & A_F & 0 \\ 0 & \bar{\sigma} D_Z 1 & 0 & 0 & 0 & 0 & E_{LF} & 0 \\ 0 & 0 & \bar{\sigma} D_Z 2 & 0 & 0 & 0 & -E_{UF} & 0 \\ 0 & 0 & 0 & \bar{\sigma} D_W 1 & 0 & L_{LF} & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{\sigma} D_W 2 & -L_{UF} & 0 & 0 \\ 0 & 0 & 0 & -L_{LF}^T & L_{UF}^T & \sigma I_F^s & 0 & L_F \\ -A_F^T & -E_{LF}^T & E_{UF}^T & 0 & 0 & 0 & H_F + \sigma I_F^x & -J_F^T \\ 0 & 0 & 0 & 0 & 0 & -L_F^T & J_F & \bar{\sigma} D_Y \end{pmatrix} \begin{pmatrix} \widetilde{q}_1 \\ \widetilde{q}_2 \\ \widetilde{q}_3 \\ \widetilde{q}_4 \\ \widetilde{q}_5 \\ \widetilde{q}_6 \\ \widetilde{q}_7 \\ \widetilde{q}_8 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ L_F r_6 \\ E_F r_7 \\ r_8 \end{pmatrix},$$

Premultiplying these equations by

$$\begin{pmatrix} I_A &&&&&&& \\ 0 & I^x_{LF} &&&&&& \\ 0 & 0 & I^x_{UF} &&&&& \\ 0 & 0 & 0 & I^s_{LF} &&&& \\ 0 & 0 & 0 & 0 & I^s_{UF} &&& \\ 0 & 0 & 0 & \frac{1}{\bar\sigma} L^T_{LF}(D_W 1)^{-1} & -\frac{1}{\bar\sigma} L^T_{UF}(D_W 2)^{-1} & I^s_F && \\ \frac{1}{\bar\sigma} A^T_F D^{-1}_A & \frac{1}{\bar\sigma} E^T_{LF}(D_Z 1)^{-1} & -\frac{1}{\bar\sigma} E^T_{UF}(D_Z 2)^{-1} & 0 & 0 & 0 & I^x_F & \\ 0 & 0 & 0 & \breve{D}_W L^T_L (D_W 1)^{-1} & -\breve{D}_W L^T_U (D_W 2)^{-1} & \bar\sigma \breve{D}_W L^T_F & 0 & I_m \end{pmatrix}$$

gives the block upper-triangular system

$$\begin{pmatrix} \bar{\sigma}D_A & 0 & 0 & 0 & 0 & 0 & A_F & 0 \\ 0 & \bar{\sigma}D_Z1 & 0 & 0 & 0 & 0 & E_{LF} & 0 \\ 0 & 0 & \bar{\sigma}D_Z2 & 0 & 0 & 0 & -E_{UF} & 0 \\ 0 & 0 & 0 & \bar{\sigma}D_W1 & 0 & L_{LF} & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{\sigma}D_W2 & -L_{UF} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\bar{\sigma}}L_F\breve{D}_W^{\dagger}L_F^T & 0 & L_F \\ 0 & 0 & 0 & 0 & 0 & 0 & \widehat{H}_F + \sigma I_F^x & -J_F^T \\ 0 & 0 & 0 & 0 & 0 & 0 & J_F & \bar{\sigma}(D_Y + \breve{D}_W) \end{pmatrix} \begin{pmatrix} \widetilde{q}_1 \\ \widetilde{q}_2 \\ \widetilde{q}_5 \\ \widetilde{q}_4 \\ \widetilde{q}_5 \\ \widetilde{q}_6 \\ \widetilde{q}_7 \\ \widetilde{q}_8 \end{pmatrix}$$

$$= \begin{pmatrix} r_1 \\ r_2 \\ r_5 \\ r_4 \\ r_5 \\ \dfrac{1}{\bar{\sigma}}L_F\left(L_L^T(D_W1)^{-1}r_4 - L_U^T(D_W2)^{-1}r_5 + \bar{\sigma}r_6\right) \\ \dfrac{1}{\bar{\sigma}}E_F\left(A^TD_A^{-1}r_1 + E_L^T(D_Z1)^{-1}r_2 - E_U^T(D_Z2)^{-1}r_3 + \bar{\sigma}r_7\right) \\ \breve{D}_W\left(L_L^T(D_W1)^{-1}r_4 - L_U^T(D_W2)^{-1}r_5 + \bar{\sigma}r_6\right) + r_8 \end{pmatrix},$$

with $\widehat{H}_F = E_F\Big(H + \frac{1}{\bar{\sigma}}\big(J^{\mathrm{T}}D_Y^{-1}J + A^{\mathrm{T}}D_A^{-1}A + D_Z^{\dagger}\big)\Big)E_F^T$. Using block back-substitution, $\widetilde{q}_7$ and $\widetilde{q}_8$ can be computed by solving the equations

$$\begin{pmatrix} \widehat{H}_F & -J_F^T \\ J_F & \bar{\sigma}\big(D_Y + \breve{D}_W\big) \end{pmatrix} \begin{pmatrix} \widetilde{q}_7 \\ \widetilde{q}_8 \end{pmatrix} = \begin{pmatrix} \frac{1}{\bar{\sigma}} E_F \left(A^{\mathrm{T}} D_A^{-1} r_1 + E_L^T (D_Z 1)^{-1} r_2 - E_U^T (D_Z 2)^{-1} r_3 + \bar{\sigma} r_7\right) \\ \breve{D}_W \left(L_L^T (D_W 1)^{-1} r_4 - L_U^T (D_W 2)^{-1} r_5 + \bar{\sigma} r_6\right) + r_8 \end{pmatrix},$$

with the remaining vectors computed as

$$\begin{aligned}
\widetilde{q}_6 &= \breve{D}_W \left(L_L^T (D_W 1)^{-1} r_4 - L_U^T (D_W 2)^{-1} r_5 + \bar{\sigma}(r_6 - \widetilde{q}_8)\right) \\
\widetilde{q}_5 &= \frac{1}{\bar{\sigma}} (D_W 2)^{-1} \big(r_5 - L_U L_F^T \widetilde{q}_6\big) \\
\widetilde{q}_4 &= \frac{1}{\bar{\sigma}} (D_W 1)^{-1} \big(r_4 - L_L L_F^T \widetilde{q}_6\big) \\
\widetilde{q}_3 &= \frac{1}{\bar{\sigma}} (D_Z 2)^{-1} \big(r_3 - E_U E_F \widetilde{q}_7\big) \\
\widetilde{q}_2 &= \frac{1}{\bar{\sigma}} (D_Z 1)^{-1} \big(r_2 - E_L E_F \widetilde{q}_7\big) \\
\widetilde{q}_1 &= \frac{1}{\bar{\sigma}} (D_A)^{-1} \big(r_1 - A_F \widetilde{q}_7\big).
\end{aligned}$$

Bibliography

[1] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK users' guide*. SIAM, 1999.

[2] Hande Y Benson and David F Shanno. "Interior-point methods for nonconvex nonlinear programming: cubic regularization". *Computational optimization and applications* 58 (2014), pp. 323–346.

[3] El Houcine Bergou, Youssef Diouane, and Serge Gratton. "A line-search algorithm inspired by the adaptive cubic regularization framework and complexity analysis". *Journal of Optimization Theory and Applications* 178.3 (2018), pp. 885–913.

[4] Tommaso Bianconcini and Marco Sciandrone. "A cubic regularization algorithm for unconstrained optimization using line search and nonmonotone techniques". *Optimization Methods and Software* 31.5 (2016), pp. 1008–1035.

[5] James R Bunch and Beresford N Parlett. "Direct methods for solving symmetric indefinite systems of linear equations". *SIAM Journal on Numerical Analysis* 8.4 (1971), pp. 639–655.

[6] Richard H Byrd, Humaid Fayez Khalfan, and Robert B Schnabel. "Analysis of a symmetric rank-one trust region method". *SIAM Journal on Optimization* 6.4 (1996), pp. 1025–1039.

[7] Coralia Cartis, Nicholas I M Gould, and Ph L Toint. "Complexity bounds for second-order optimality in unconstrained optimization". *Journal of Complexity* 28.1 (2012), pp. 93–108.

[8] Coralia Cartis, Nicholas I M Gould, and Philippe L Toint. "Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results". *Mathematical Programming* 127.2 (2011), pp. 245–295.

[9] Coralia Cartis, Nicholas I M Gould, and Philippe L Toint. "Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity". *Mathematical programming* 130.2 (2011), pp. 295–319.

[10] Coralia Cartis, Nicholas I M Gould, and Philippe L Toint. "Trust-region and other regularisations of linear least-squares problems". *BIT Numerical Mathematics* 49.1 (2009), pp. 21–53.

[11] Alan K Cline, Cleve B Moler, George W Stewart, and James H Wilkinson. "An estimate for the condition number of a matrix". *SIAM Journal on Numerical Analysis* 16.2 (1979), pp. 368–375.

[12] Andrew Conn, Nick Gould, and Ph Toint. "A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds". *Mathematics of Computation* 66.217 (1997), pp. 261–288.

[13] Andrew R Conn, Nicholas I M Gould, and Ph L Toint. "Convergence of quasi-Newton matrices generated by the symmetric rank one update". *Mathematical programming* 50.1 (1991), pp. 177–195.

[14] Andrew R Conn, Nicholas I M Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.

[15] T Dehghan Niri, M Heydari, and M M Hosseini. "Two modified adaptive cubic regularization algorithms by using the nonmonotone Armijo-type line search". *Optimization* (2022), pp. 1–24.

[16] Elizabeth D Dolan and Jorge J Moré. "Benchmarking optimization software with performance profiles". *Mathematical programming* 91 (2002), pp. 201–213.

[17] Anders Forsgren. "Inertia-controlling factorizations for optimization algorithms". *Applied Numerical Mathematics* 43.1-2 (2002), pp. 91–107.

[18] Anders Forsgren and Philip E Gill. "Primal-dual interior methods for nonconvex nonlinear programming". *SIAM Journal on Optimization* 8.4 (1998), pp. 1132–1152.

[19] Anders Forsgren, Philip E Gill, and Margaret H Wright. "Interior methods for nonlinear optimization". *SIAM review* 44.4 (2002), pp. 525–597.

[20] David M Gay. "Computing optimal locally constrained steps". *SIAM Journal on Scientific and Statistical Computing* 2.2 (1981), pp. 186–197.

[21] E Michael Gertz and Philip E Gill. "A primal-dual trust region algorithm for nonlinear optimization". *Math. Program* 100.1 (2002).

[22] Philip E Gill, Vyacheslav Kungurtsev, and Daniel P Robinson. "A shifted primal-dual penalty-barrier method for nonlinear optimization". *SIAM Journal on Optimization* 30.2 (2020), pp. 1067–1093.

[23] Philip E Gill, Walter Murray, and Michael A Saunders. "SNOPT: An SQP algorithm for large-scale constrained optimization". *SIAM review* 47.1 (2005), pp. 99–131.

[24] Donald Goldfarb, R Polyak, Katya Scheinberg, and I Yuzefovich. "A modified barrier-augmented Lagrangian method for constrained minimization". *Computational optimization and applications* 14 (1999), pp. 55–74.

[25] Nicholas I M Gould, Dominique Orban, and Philippe L Toint. "CUTEr and SifDec: A constrained and unconstrained testing environment, revisited". *ACM Transactions on Mathematical Software (TOMS)* 29.4 (2003), pp. 373–394.

[26] Nicholas I M Gould, Dominique Orban, and Philippe L Toint. "CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization". *Computational optimization and applications* 60 (2015), pp. 545–557.

[27] Nicholas I M Gould, Margherita Porcelli, and Philippe L Toint. "Updating the regularization parameter in the adaptive cubic regularization algorithm". *Computational optimization and applications* 53.1 (2012), pp. 1–22.

[28] Nicholas I M Gould, Daniel P Robinson, and H Sue Thorne. "On solving trust-region and other regularised subproblems in optimization". *Mathematical Programming Computation* 2.1 (2010), pp. 21–57.

[29] Andreas Griewank. *The modification of Newton's method for unconstrained optimization by bounding cubic terms.* Tech. rep. Technical report NA/12, 1981.

[30] Luigi Grippo, Francesco Lampariello, and Stephano Lucidi. "A nonmonotone line search technique for Newton's method". *SIAM Journal on Numerical Analysis* 23.4 (1986), pp. 707–716.

[31] William W Hager. "Condition estimates". *SIAM Journal on scientific and statistical computing* 5.2 (1984), pp. 311–316.

[32] Nicholas J Higham. "FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation". *ACM Transactions on Mathematical Software (TOMS)* 14.4 (1988), pp. 381–396.

[33] E Michael Gertz. "A quasi-Newton trust-region method". *Mathematical Programming* 100.3 (2004), pp. 447–470.

[34] Jorge J Moré and Danny C Sorensen. "Computing a trust region step". *SIAM Journal on Scientific and Statistical Computing* 4.3 (1983), pp. 553–572.

[35] Stephen G Nash, R Polyak, and Ariela Sofer. "A numerical comparison of barrier and modified barrier methods for large-scale bound-constrained optimization". *Large scale optimization: State of the Art* (1994), pp. 319–338.

[36] Yurii Nesterov and Boris T Polyak. "Cubic regularization of Newton method and its global performance". *Mathematical Programming* 108.1 (2006), pp. 177–205.

[37] Roman Polyak. “Modified barrier functions (theory and methods)”. *Mathematical programming* 54 (1992), pp. 177–222.

[38] Stephen Wright and Jorge Nocedal. “Numerical Optimization”. *Springer Science* 35.67-68 (1999), p. 7.

[39] Hongchao Zhang and William W Hager. “A nonmonotone line search technique and its application to unconstrained optimization”. *SIAM Journal on Optimization* 14.4 (2004), pp. 1043–1056.