UNIVERSITY OF CALIFORNIA SAN DIEGO

Primal-Dual Path-Following Methods For Nonlinear Programming

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

in

Mathematics

by

Fangyao Su

Committee in charge:

Professor Philip E. Gill, Chair
Professor Randolph E. Bank
Professor Thomas R. Bewley
Professor Hyunsun A. Kim
Professor Jiawang Nie

2019

The Dissertation of Fangyao Su is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

EPIGRAPH

I care not whether I can achieve,
for the longed goal,
I shall despite wind and rain go.
—*Guozhen Wang*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGEMENTS

First of all, I would like to express the deepest gratitude to my advisor, Professor Philip E. Gill for his constant encouragement and guidance, for passing his rigorous attitude as a mathematician to me, and for his knowledge, patience and support for me in all aspects over the past 5 years. As a world-leading expert in optimization, he leads me into this branch of mathematics that used to be totally fresh to me.

I would like to acknowledge Profs. Randolph E. Bank, Jiawang Nie for the knowledge they taught me in graduate classes at UCSD and Profs. Thomas R. Bewley, Hyunsun A. Kim for their time to serve as my committee member.

I am very grateful for Yuanjie Jiang for her love and encouragement, my parents for their unconditional support to my career and life. I would also like to thank my friends Dun Qiu, Xiudi Tang, Xuefeng Shen, Yi Luo, Zihao Li and David Lenz at UCSD for their discussions with me on math problems and my officemates Yesheng Huang, Yingjia Fu and Yucheng Tu for their help in all aspects.

Chapter 3, in part is currently being prepared for submission for publication of the material. Su, Fangyao; Gill, Philip E. The dissertation author was the primary investigator and author of this material.

Chapter 5, in part is currently being prepared for submission for publication of the material. Su, Fangyao; Gill, Philip E. The dissertation author was the primary investigator and author of this material.

VITA

2014	Bachelor of Science in Mathematics University of Science and Technology of China
2015-2019	Teaching Assistant University of California San Diego
2017-2018	Research Assistant University of California San Diego
2019	Doctor of Philosophy in Mathematics University of California San Diego

ABSTRACT OF THE DISSERTATION

Primal-Dual Path-Following Methods For Nonlinear Programming

by

Fangyao Su

Doctor of Philosophy in Mathematics

University of California San Diego, 2019

Professor Philip E. Gill, Chair

The main goal of this dissertation is to study the formulation and analysis of primal-dual path-following methods for nonlinear programming (NLP), which involves the minimization or maximization of a nonlinear objective function subject to constraints on the variables. Two important types of nonlinear program are problems with nonlinear equality constraints and problems with nonlinear inequality constraints. In this dissertation, two new methods are proposed for nonlinear programming. The first is a new primal-dual path-following augmented Lagrangian method (PDAL) for solving a nonlinear program with equality constraints only. The second is a new primal-dual path-following shifted penalty-barrier method (PDPB) for solving a nonlinear program with a mixture of equality and inequality constraints. The method of PDPB may be regarded as an extension of PDAL to handle nonlinear inequality constraints.

Algorithms PDAL and PDPB are iterative methods that share the same “two-level” structure involving outer and inner iterations. In the outer iteration of PDAL, the optimality conditions are perturbed to define a "path-following trajectory" parameterized by a set of Lagrange multiplier estimates and a penalty parameter. The iterates are constructed to closely follow the trajectory towards a constrained local minimizer of the nonlinear program. If an outer iterate deviates significantly from the trajectory, then an inner iteration is invoked in which a primal-dual augmented Lagrangian merit function is minimized to force the iterates back to a neighborhood of the trajectory.

A similar approach is used to handle the inequality constraints in PDPB. In this case, the trajectory is followed towards a local solution of the mixed-constraint nonlinear program. This trajectory is parameterized by a set of Lagrange multiplier estimates and penalty and barrier parameters associated with the equality and inequality constraints. If an iterate moves away from the trajectory, a primal-dual shifted penalty-barrier merit function is minimized using a trust-region method. By introducing slack variables, global convergence can be achieved from any starting point without the need for an initial strictly feasible point. Furthermore, numerical experiments indicate that when minimizing the shifted barrier function, the trust-region method requires fewer matrix factorizations and iterations than a comparable line-search method.

Chapter 1

Introduction

1.1 Problem Description

In constrained optimization problems, the optimal value of objective function and its corresponding solution set must be found subject to certain constraints being satisfied. One of the most general formats used to express a nonlinear program is given by

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \\ &\text{subject to} \;\; c_\ell \leq \; c(x) \leq c_u, \quad x_\ell \leq \; x \leq x_u, \end{aligned} \tag{1.1}$$

where $f(x) : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function and $c(x) : \mathbb{R}^n \mapsto \mathbb{R}^m$ are the constraints. c_ℓ and c_u are the lower and upper bounds of $c(x)$ in $\mathbb{R}^m$, x_ℓ and x_u are the lower and upper bounds of x in $\mathbb{R}^n$. As maximizing $f(x)$ is equivalent to minimizing $-f(x)$, without loss of generality, only minimizing an objective function will be considered.

Problem (1.1) can be written in the following simpler equivalent format in which each constraint is expressed in terms of a single nonnegative function:

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

Problem (NIP) is the main focus of this dissertation.

Given a vector s of nonnegative slack variables, problem (NIP) can be written in the equivalent form

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

A special case of the problems (NIP) and (NIPs) occurs when all the constraints are equalities. In this case the problem is

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

In general, problem (NEP) is easier to solve than problem (NIP) and has a number of useful properties. Chapter 3 starts with a proposed primal-dual path-following augmented Lagrangian method for problem (NEP) and then this method is extended with the barrier term introduced to deal with the inequality constraints in Chapter 5.

Common types of optimization problem include linear programming (LP), where both $f(x)$ and $c(x)$ are affine functions, and quadratic programming (QP) where $f(x)$ is in quadratic form and $c(x)$ is affine. Optimization problems may also be categorized in terms of the level of smoothness of the objective and constraint functions. This dissertation concerns the formulation and analysis of algorithms for problems in which $f(x)$ and $c(x)$ are *twice continuously differentiable*, *nonlinear* and *nonconvex*. In all cases it is assumed that the gradient and Hessian are available at any given point. Knowledge of the second derivatives allows the formulation of methods with a superlinear or quadratic rate of convergence to a solution.

Finally, only algorithms for finding *local minimizers* are considered in this dissertation.

1.2 Contributions of This Dissertation

In this dissertation, Chapter 2 serves as a general background for all the subsequent chapters. In Chapter 3, a primal-dual path-following augmented Lagrangian method (PDAL) is proposed for solving problem (NEP). In this method, the optimality conditions are perturbed to define a path-following trajectory, and all the iterates are forced to follow it towards a local constrained minimizer of problem (NEP). Once some iterate departs from this trajectory, an augmented Lagrangian type merit function that measures the distance between current iterate and the trajectory will be minimized using line-search method.

Like many other primal-dual methods proposed by Gill and coauthors (see [10, 12, 18, 21]), the proposed algorithms treat both the primal and dual variables as independent variables at each iteration. This strategy is different from that of other methods that regard the dual variables as being dependent on the primal variables and update them each time the primal variables are modified. This can provide certain benefits, such as the ability to control the quality of the dual variables during the solution of each subproblem, and the ability to impose explicit bounds on the dual variables. These strategies can improve both the efficiency and the reliability of a method.

In Chapter 5, the method of PDAL is extended to solve problem (NIP). In this chapter, a new primal-dual path-following shifted penalty-barrier method (PDPB) is proposed with some features inherited from PDAL. In the inequality constrained case the algorithm follows a trajectory defined by perturbing the optimality conditions of problem (NIP). If an iterate departs from the trajectory, a modified penalty-barrier function is minimized using a combined trust-region line-search method. As in the case of algorithm PDAL, this strategy guarantees global convergence. Unlike PDAL where line-search techniques are used, PDPB is a trust-region method, which implies that unlike a line-search method, it is not necessary to use an inertia-controlling factorization of the Hessian of the Lagrangian function. Furthermore, it can be shown that when minimizing the shifted-barrier function, incorporating a line-search technique into the trust-region method generally requires fewer matrix factorizations and fewer iterations, which brings certain numerical benefits.

1.3 Notation

Most of notations in this dissertation are consistent with the standard optimization literature. $f(x)$ specifically refers to the objective function with $g(x)$ and $H(x)$ being its gradient and Hessian. $c(x)$ specifically refers to the constraint function with $J(x)$ being its Jacobian matrix, whose i-th row is defined by $\nabla c_i(x)^T$. The subscript k means the function takes value at iterate x_k, say, g_k means $g(x_k)$. I refers to identity matrix with its dimension inferred from the context. e is a vector of all ones and e_i is the i-th column of identity matrix. For a symmetric matrix A, the inertia of A, denoted by $\mathrm{In}(A)$, is the 3-tuple (i_+, i_-, i_0) indicating the number of positive, negative and zero eigenvalues of A. The norm $||\cdot||$ refers to vector 2-norm and its induced matrix 2-norm unless otherwise stated. $x \cdot y$ is the element-wise product of two column vectors such that $[x \cdot y]_i = x_i y_i$, similarly, $\min(x, y)$ is a vector with components $\min(x_i, y_i)$. The vector consisting of x augmented by y is defined by (x, y). Given $\delta > 0$, an *open ball* at x^* is denoted by $\mathcal{B}(x^*, \delta) = \{x : ||x - x^*|| < \delta\}$. The i-th eigenvalue of a matrix A is denoted by $\lambda_i(A)$. Finally, $H \succeq 0$ and $H \succ 0$ refer to matrix H being positive semidefinite and positive definite respectively. The $\{H_k\}_{k\geq 0}$ are said to be uniformly positive definite if there exists some constant $\lambda > 0$ such that for all $k \in \mathbb{N}$ and all $p \in \mathbb{R}^n$, $p^T H_k p \geq \lambda ||p||^2$. If there exists a positive constant γ such that $||\alpha_j|| \leq \gamma\beta_j$, then write $\alpha_j = O(\beta_j)$. If there exists a sequence $\{\gamma_j\}_{j\geq 0} \to 0$ such that $||\alpha_j|| \leq \gamma_j\beta_j$, then write $\alpha_j = o(\beta_j)$.

1.4 Some Useful Results

In this section, some useful results will be described that will be used frequently without proofs throughout this dissertation.

Lemma 1.4.1 (Sylvester's law of inertia)**.** *If K is symmetric, then* $\mathrm{In}(X^TKX) = \mathrm{In}(K)$ *for any nonsingular matrix X, where* $\mathrm{In}(K)$ *denotes the inertia of K, i.e., the number of positive, negative and zero eigenvalues of K.* □

Lemma 1.4.2 (Inertia of the KKT matrix I)**.** *Given an $n \times n$ symmetric matrix H and an $m \times n$ matrix J, let r denote the rank of J and let $Z \in \mathbb{R}^{n\times m}$ be a matrix whose columns span the null space of J. Consider a Karush–Kuhn–Tucker (KKT) matrix of the form*

$$K = \begin{pmatrix} H & J^T \\ J & 0 \end{pmatrix}.$$

Then the inertia of K is given by

$$\mathrm{In}(K) = \mathrm{In}(Z^THZ) + (r, r, m - r).$$

The matrix Z^THZ is known as the reduced Hessian and describes the curvature of objective function on the active constraint surface. □

Lemma 1.4.3 (Inertia of the KKT matrix II)**.** *Given an $n \times n$ symmetric matrix H and an $m \times n$ matrix J of rank m, let $Z \in \mathbb{R}^{n\times m}$ be a matrix whose columns span the null space of J. If a KKT matrix K is given by*

$$K = \begin{pmatrix} H & J^T \\ J & 0 \end{pmatrix},$$

then the inertia of K is given by

$$\mathrm{In}(K) = \mathrm{In}(Z^THZ) + (m, m, 0).$$

If Z^THZ is positive definite, then $\mathrm{In}(K) = (n, m, 0)$*, in which case, the KKT matrix K is said to have the correct inertia.* □

Proof. Set $r = m$ in Lemma 1.4.2. □

Lemma 1.4.4 (Inertia of the KKT matrix III)**.** *Given an $n \times n$ symmetric matrix H and an $m \times n$ matrix J of rank r, let the $n - r$ columns of Z define a basis for null(J) and define the regularized KKT matrix as*

$$K_\mu = \begin{pmatrix} H & J^T \\ J & -\mu I \end{pmatrix}. \tag{1.2}$$

If $\mu > 0$ is sufficiently small, then the following statements are equivalent.

- *$H + \frac{1}{\mu} J^T J$ is positive definite.*
- *$\mathrm{In}(K_\mu) = (n, m, 0)$.*
- *$Z^T H Z$ is positive definite.* □

Proof. Use the Schur complement and Debreu's Lemma, which is given below. □

Notice that in Lemma 1.4.4, there is no need for J to have full rank anymore, as long as $\mu > 0$ sufficiently small and the reduced Hessian is positive definite, K_μ will have the correct inertia. The $-\mu I$ term in (2,2) block is generally regarded as a *regularization* of the KKT matrix.

Lemma 1.4.5 (Debreu's Lemma)**.** *There exists a finite $\bar{\rho} > 0$ such that $H + \rho J^T J$ is positive definite for all $\rho \geq \bar{\rho}$ if and only if $p^T H p > 0$ for all nonzero p such that $Jp = 0$.* □

The matrix K_μ in (1.2) represents a "general" form of KKT matrix, where $(1, 1)$ block represents the Hessian of Lagrangian function or its approximation and $(1, 2)$ block represents the Jacobian of the constraints with (2,2) block being 0 or some diagonal matrix. In the context of the augmented Lagrangian method or shifted barrier method, KKT matrix is generally much more complicated but shares the same structure with K_μ. Moreover, the techniques used to modify the inertia of matrix in the form of K_μ to solve the KKT system will also be frequently used in this dissertation.

Chapter 2

Background

This chapter provides some background results for the subsequent chapters. Section 2.1 introduces the optimality conditions and corresponding constraint qualifications for both (NEP) and (NIP). Section 2.2 describes Newton's method and the commonly used line-search strategies. The background of a conventional augmented Lagrangian method for (NEP), which is the basis of the proposed primal-dual path-following augmented Lagrangian method, is considered in Chapter 3. Similarly, the first part of Chapter 5 describes the conventional barrier method for (NIP) that serves as a basis for the proposed primal-dual path-following shifted penalty-barrier method. Finally, the background theory of the combined trust-region line-search method used to minimize the primal-dual shifted penalty-barrier function is discussed in Chapter 4.

2.1 Optimality Conditions

2.1.1 Optimality Conditions for (NEP)

In this section the constraint qualifications and optimality conditions for a constrained local minimizer are provided for problem (NEP). First, the definition of a constrained local minimizer is given.

Definition 2.1.1 (Local minimizer). *Let $f(x) : \mathbb{R}^n \mapsto \mathbb{R}$ be the objective function and let $\mathcal{F} = \{x : c(x) = 0\}$ denote the feasible region. A point x^* is a constrained local minimizer of $f(x)$ if $x^* \in \mathcal{F}$ and there exists an open ball $\mathcal{B}(x^*, \delta)$ such that*

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

Furthermore, x^* is a strict constrained local minimizer if

$$f(x^*) < f(x) \textit{ for all } x \in \mathcal{B}(x^*, \delta) \cap \mathcal{F}, \quad x \neq x^*.$$

The point x^* is an isolated constrained minimizer if there exists a positive δ such that x^* is the unique constrained local minimizer in $\mathcal{B}(x^*, \delta) \cap \mathcal{F}$. □

Definition 2.1.2 (Feasible path). *Assume that x is a feasible point, i.e., $c(x) = 0$. A feasible path is a twice continuously differentiable curve $x(\alpha)$ such that*

- *$x(0) = x$, $c(x(\alpha)) = 0$ for all $0 \leq \alpha < \hat{\alpha}$ and some $\hat{\alpha} > 0$; and*
- *the tangent vector $\frac{d}{d\alpha}x(\alpha)$ is nonzero at $\alpha = 0$.* □

Definition 2.1.3 (Level set and Level curves). *Given a function $f(x) : \mathcal{D} \in \mathbb{R}^n \mapsto \mathbb{R}$ defined on a convex set $\mathcal{D}$, the level set $\mathcal{L}(\gamma)$ associated with the scalar γ, is the set*

$$\mathcal{L}(\gamma) = \{x \in \mathcal{D} : f(x) \leq \gamma\}.$$

The boundary of the level set $\mathcal{L}(\gamma)$, i.e., $\{x \in \mathcal{D} : f(x) = \gamma\}$ is called the level curve associated with γ. □

Definition 2.1.4 (Lagrange multipliers). *Assume that x^* is a constrained local minimizer for (NEP), then the components of the vector y^* such that $g(x^*) = J(x^*)^T y^*$ are called Lagrange multipliers.* □

Sometimes the Lagrange multipliers are called the *dual variables*, with the terminology initially coming from linear programming. The Lagrange multipliers are essential in indicating whether the primal variable x is optimal. However, in order to use y^* to indicate the optimality, certain regularity conditions or constraint qualifications must be described first.

Definition 2.1.5 (General constraint qualification)**.** *The constraint qualification for* $c(x) = 0$ *holds at* x *if every* $p \neq 0$ *such that* $J(x)p = 0$ *is tangential to a differentiable feasible path starting at* x*.* □

It is now possible to state the first-order optimality conditions for (NEP).

Theorem 2.1.1 (First-order optimality condition)**.** *If the constraint qualification holds at* x^**, then* x^* *is a local solution of (NEP) only if there exist Lagrange multipliers* y^**, such that the following condition holds*

$$g(x^*) = J(x^*)^T y^* = \sum_{i=1}^{m} y_i^* \nabla c_i(x^*),$$

or, equivalently, if the columns of $Z(x^*)$ *form a basis for the null-space of* $J(x^*)$*, then* $Z(x^*)^T g(x^*) = 0$*. The vector* $Z(x)^T g(x)$ *is known as the reduced gradient.* □

A point that satisfies the first-order optimality condition is called a *first-order KKT point* of (NEP). The goal is to find constrained local *minimizers* of the objective function, and to distinguish between constrained saddle points, local maximizers and local minimizers. This means that the curvature of $f(x)$ on the constraint surface must be considered. To do so, it is convenient to define the Lagrangian function as follows.

Definition 2.1.6 (Lagrangian function)**.** *Given Lagrange multipliers* y*, the Lagrangian function* $L : \mathbb{R}^{n+m} \mapsto \mathbb{R}$ *is defined by*

$$L(x, y) = f(x) - y^T c(x).$$

The second-order derivative of $L(x, y)$ with respect to x describes the curvature of objective function $f(x)$ on the constraint surface, and is given by

$$H(x,y) = \nabla^2_{xx} L(x,y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 c_i(x).$$

The gradient and Hessian of Lagrangian function can be expressed as

$$\nabla L(x,y) = \begin{pmatrix} g(x) - J(x)^T y \\ -c(x) \end{pmatrix} \qquad \nabla^2 L(x,y) = \begin{pmatrix} H(x,y) & -J(x)^T \\ -J(x) & 0 \end{pmatrix}.$$

Now it is possible to define the second-order optimality conditions for (NEP).

Theorem 2.1.2 (Second-order optimality condition)**.** *If the constraint qualification holds at x^*, then x^* is a local solution of (NEP) only if the following conditions hold.*

- *x^* is feasible, i.e., $c(x^*) = 0$;*
- *there exist Lagrange multipliers y^* such that $g(x^*) = J(x^*)^T y^*$; and*
- *for the y^* above, $p^T H(x^*, y^*)p \geq 0$ for all p such that $J(x^*)p = 0$.* □

A point that satisfies the second-order optimality condition is called a *second-order KKT point* for (NEP). A compact statement of the last condition in Theorem 2.1.2 can be described by the *reduced Hessian* $Z(x^*)^T H(x^*, y^*) Z(x^*)$ being positive semidefinite, where columns of $Z(x^*)$ form a basis of null-space of $J(x^*)$.

The second-order *sufficient* condition for determining a *strict* local minimizer or furthermore, an *isolated* local minimizer of (NEP) can be summarized in the following two theorems, where no assumption on constraint qualifications is needed.

Theorem 2.1.3 (Second-order sufficient condition I)**.** *A point x^* is a strict local minimizer of problem (NEP) if the following conditions hold.*

- *x^* is feasible, i.e., $c(x^*) = 0$;*
- *there exist Lagrange multipliers y^* such that $g(x^*) = J(x^*)^T y^*$; and*

- *for the y^* above, it holds that $p^T H(x^*, y^*)p > 0$ for all $p \neq 0$ such that $J(x^*)p = 0$, or equivalently, the reduced Hessian $Z(x^*)^T H(x^*, y^*)Z(x^*)$ is positive definite.* □

Theorem 2.1.4 (Second-order sufficient condition II)**.** *A point x^* is an isolated local minimizer of (NEP) if the following conditions hold.*

- *x^* is feasible, i.e., $c(x^*) = 0$;*
- *there exists Lagrange multipliers y^* such that $g(x^*) = J(x^*)^T y^*$;*
- *for the y^* above, it holds that $p^T H(x^*, y^*)p > 0$ for all $p \neq 0$ such that $J(x^*)p = 0$, or, equivalently, the reduced Hessian $Z(x^*)^T H(x^*, y^*)Z(x^*)$ is positive definite; and*
- *the constraint gradients are linearly independent at x^*.* □

Note that the third condition in both Theorem 2.1.3 and Theorem 2.1.4 requires the reduced Hessian to be positive definite, so the curvature of objective function $f(x)$ on the constraint surface must be bounded away from zero.

2.1.2 Optimality Conditions for (NIP)

In this part, the constraint qualifications and optimality conditions of a constrained local minimizer will be provided for problem (NIP). These conditions are used frequently Chapter 5. As before, certain constraint qualifications are needed to imply that the KKT conditions are necessary conditions for some x^* to be a first-order solution. Commonly used constraint qualifications include *linear independence constraint qualification (LICQ)*, *Mangasarian-Fromovitz constraint qualification (MFCQ)*, and *Slater constraint qualification*.

Definition 2.1.7 (LICQ)**.** *Denote the active set $\mathcal{A}(x) = \{i : c_i(x) = 0\}$, LICQ holds at x^* if the active constraint gradients, $\{\nabla c_i(x^*) : i \in \mathcal{A}(x^*)\}$ are linearly independent, i.e., $J_a(x^*)$ has full rank.* □

Definition 2.1.8 (MFCQ)**.** *Denote the active set $\mathcal{A}(x) = \{i : c_i(x) = 0\}$, MFCQ holds at x^* if there exists an "interior" vector p starting at x^*, i.e., there exists a vector p such that $\nabla c_i(x)^T p > 0$ for all $i \in \mathcal{A}(x^*)$, i.e., $J_a(x^*)p > 0$.* □

Definition 2.1.9 (Slater CQ)**.** *The Slater constraint qualification holds if the set $\{-c_i(x)\}$ is convex and there exists an "interior" feasible point, i.e., there exists $\widehat{x}$ such that $c_i(\widehat{x}) > 0$ for every index i.* □

Generally speaking, LICQ is computationally tractable and more practical than MFCQ, although it is a stronger condition than MFCQ. If LICQ holds at some KKT point x^*, then the Lagrange multipliers are unique because $J_a(x^*)$ has full rank. On the other hand, the Slater constraint qualification is fairly weak in the sense that it only requires that a strictly feasible (i.e., interior) point exists.

Definition 2.1.10 (First-order KKT point)**.** *The first-order KKT conditions of (NIP) hold at x^* if there exist Lagrange multipliers y^* satisfying*

- $c(x^*) \geq 0$ *(Feasibility)*
- $g(x^*) = J(x^*)^T y^*$ *(Stationarity)*
- $y^* \geq 0$ *(Nonnegativity of the multipliers)*
- $c(x^*) \cdot y^* = 0$ *(Complementarity)* □

Definition 2.1.11 (CAKKT point)**.** *In solving problem (NIP) with the slack variables s introduced in the following form*

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

A point (x^*, s^*) satisfying $s^* \geq 0$ and $c(x^*) - s^* = 0$ is said to satisfy the CAKKT condition if there exists a sequence $\{(x_k, s_k, y_k, w_k)\}$ with $\{x_k\} \to x^*$ and $\{s_k\} \to s^*$ such that

$$\begin{gathered} \lim_{k\to\infty} \big(g(x_k) - J(x_k)^T y_k\big) = 0, \quad \lim_{k\to\infty} (y_k - w_k) = 0, \\ \lim_{k\to\infty} (s_k \cdot w_k) = 0, \quad w_k \geq 0. \end{gathered} \tag{2.1}$$

Any pair (x^*, s^*) satisfying the CAKKT condition is called a CAKKT point. □

CAKKT point is useful by the following Lemma 2.1.5, which uses *cone-continuity property*, the weakest constraint qualification associated with sequential optimality conditions (see Andreani et al. [1]).

Lemma 2.1.5. *If (x^*, s^*) is a CAKKT point that satisfies the cone-continuity property, then (x^*, s^*) is a first-order KKT point for problem (NIPs).* □

As before, the KKT conditions alone cannot distinguish between local maximizers, local minimizers and saddle points unless (NIP) is strictly convex. As before, certain second-order optimality conditions are needed. It is more complicated that when solving (NIP), second-order constraint qualifications are also needed.

Definition 2.1.12 (SOCQ)**.** *Define the second-order feasible directions $\mathcal{S}_L(x^*)$ as*

$$\mathcal{S}_L(x^*) = \left\{p : p \neq 0,\ \ g(x^*)^T p = 0 \ \ \textit{and} \ \ J_a(x^*)p \geq 0\right\}.$$

The second-order constraint qualification (SOCQ) holds at a KKT point x^* if every $p \in \mathcal{S}_L(x^*)$ is tangential to a twice-differentiable path $x(\alpha)$ such that $c_a(x(\alpha)) \geq 0$ for all $0 < \alpha \leq \hat{\alpha}$, with some $\hat{\alpha} > 0$. □

Note that the first and second order constraint qualifications are distinct assumptions, with neither implying the other. Now it is possible to describe the second-order necessary and sufficient conditions of (NIP).

Theorem 2.1.6. *Suppose both the first and second order constraint qualifications hold at a feasible point x^*. Then x^* is a local minimizer of (NIP) only if*

- *x^* is a KKT point, i.e., $c(x^*) \geq 0$ and there exists a nonempty set $\mathcal{Y}(x^*)$ of multipliers y^* satisfying $y^* \geq 0$, $c(x^*) \cdot y^* = 0$ and $g(x^*) = J(x^*)^T y^*$; and*
- *for some $y \in \mathcal{Y}(x^*)$ and all nonzero p satisfying $g(x^*)^T p = 0$ and $J(x^*)p \geq 0$, it holds that $p^T H(x^*, y^*)p \geq 0$.* □

Theorem 2.1.7. *The point x^* is a strict local minimizer of (NIP) if*

- *x^* is a KKT point, i.e., $c(x^*) \geq 0$ and there exists a nonempty set $\mathcal{Y}(x^*)$ of multipliers y^* satisfying $y^* \geq 0$, $c(x^*) \cdot y^* = 0$ and $g(x^*) = J(x^*)^T y^*$; and*
- *for some $y \in \mathcal{Y}(x^*)$ and all nonzero p satisfying $g(x^*)^T p = 0$ and $J(x^*)p \geq 0$, there exists a constant $\omega > 0$ such that $p^T H(x^*, y^*)p \geq \omega ||p||^2$.* □

As before, there are no any requirements on the constraint qualifications in order for the sufficient conditions to hold. Two more stronger sufficient conditions indicating the *isolated* local minimizers of (NIP) using either MFCQ or LICQ can be summarized in the following two theorems.

Theorem 2.1.8. *The point x^* is an isolated local minimizer of (NIP) if*

- *x^* is a KKT point, i.e., $c(x^*) \geq 0$ and there exists a nonempty set $\mathcal{Y}(x^*)$ of multipliers y^* satisfying $y^* \geq 0$, $c(x^*) \cdot y^* = 0$ and $g(x^*) = J(x^*)^T y^*$;*
- *the MFCQ holds at x^*, i.e., there exists a vector p such that $J_a(x^*)p > 0$; and*
- *for all $y \in \mathcal{Y}(x^*)$ and all nonzero p satisfying $g(x^*)^T p = 0$ and $J(x^*)p \geq 0$, there exists some $\omega > 0$ such that $p^T H(x^*, y^*)p \geq \omega ||p||^2$.* □

Theorem 2.1.9. *The point x^* is a isolated local minimizer of (NIP) if*

- *x^* is a KKT point and strict complementarity condition holds, i.e., the unique Lagrange multipliers y^* has the property that $y_i^* > 0$ for all $i \in \mathcal{A}(x^*)$;*
- *the LICQ holds at x^*, i.e., $J_a(x^*)$ has full row rank; and*
- *for every p satisfying $J_a(x^*)p = 0$, there exists some $\omega > 0$ such that $p^T H(x^*, y^*)p \geq \omega ||p||^2$.* □

Theorem 2.1.9 is more restrictive than Theorem 2.1.8 but is more computational intractable, the rank of $J_a(x^*)$ can be obtained by using the singular value decomposition on it and the strict complementarity condition on y_a^* can be examined by detecting if they are sufficiently positive. The second-order sufficient conditions motivate the definition of a second-order KKT point defined below.

Definition 2.1.13 (Second-order KKT point)**.** *A point x^* is a second-order KKT point if there exist Lagrange multipliers y^*, such that*

- $c(x^*) \geq 0, \quad g(x^*) = J(x^*)^T y^*, \quad y^* \geq 0, \quad c(x^*) \cdot y^* = 0$, *and*
- $Z(x^*)^T H(x^*, y^*) Z(x^*)$ *is positive semidefinite,*

where the columns of $Z(x^*)$ form a basis for the null space of $J(x^*)$. □

In order to simplify the description of the proposed primal-dual path-following shifted penalty-barrier method in Chapter 5, both the necessary and sufficient conditions for problem (NEIP) will also be presented.

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c_{\mathcal{E}}(x) = 0, \;\; c_{\mathcal{I}}(x) \geq 0, \tag{NEIP}$$

where $\mathcal{E}$ and $\mathcal{I}$ are nonintersecting index sets, representing the equality and inequality components of the nonlinear constraints respectively. Similarly, denote $y_{\mathcal{E}}$ and $y_{\mathcal{I}}$ to be the Lagrange multipliers associated with $c_{\mathcal{E}}$ and $c_{\mathcal{I}}$. The necessary and sufficient conditions for some point x^* to be a solution of (NEIP) can be formally described in the following two theorems.

Theorem 2.1.10 (First and second order necessary conditions)**.** *If x^* is a local minimizer of problem (NEIP) at which the MFCQ holds, then*

- *x^* is a KKT point, i.e., $c_{\mathcal{E}}(x) = 0, c_{\mathcal{I}}(x) \geq 0$ and there exists a nonempty set $\mathcal{Y}(x^*)$ of multipliers y satisfying $y_{\mathcal{I}} \geq 0, c_{\mathcal{I}}(x^*) \cdot y_{\mathcal{I}} = 0$ and $g(x^*) = J(x^*)^T y$; and*
- *every $y \in \mathcal{Y}(x^*)$ defined in the previous condition satisfies $p^T H(x^*, y^*) p \geq 0$ for all p such that $g(x^*)^T p = 0, J_{\mathcal{E}}(x^*) p = 0$ and $J_a(x^*) p \geq 0$.* □

Theorem 2.1.11 (Sufficient conditions for an isolated minimizer). *A point x^* is an isolated local minimizer of problem (NEIP) if*

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

The primal-dual path-following shifted penalty-barrier method proposed in Chapter 5 is designed to solve problems in the form of (NEIP) but with $c_{\mathcal{I}}(x) = x$. Finally, the following definitions are used to characterize the rate of convergence of a sequence $\{x_k\}_{k\geq 0}$ to x^*.

Definition 2.1.14 (Q-order convergence). *The sequence $\{x_k\}_{k\geq 0}$ is said to converge to x^* with "Q-order at least r" $(r \geq 1)$ if there exists constants $\beta \geq 0$ and $K \geq 0$ such that*

$$||x_{k+1} - x^*|| \leq \beta ||x_k - x^*||^r \text{ for all } k \geq K,$$

i.e., $||x_{k+1} - x^*|| = O(||x_k - x^*||^r)$. In some special case, convergence with Q-order at least 2 is called Q-quadratic convergence. □

Definition 2.1.15 (Q-superorder convergence). *$\{x_k\}_{k\geq 0}$ is said to converge to x^* with "Q-superorder at least r" $(r \geq 1)$ if there exists a sequence of positive constants $\{\beta_k\}_{k\geq 0}$ converging to zero and some constant $K \geq 0$ such that*

$$||x_{k+1} - x^*|| \leq \beta_k ||x_k - x^*||^r \text{ for all } k \geq K,$$

i.e., $||x_{k+1} - x^*|| = o(||x_k - x^*||^r)$. In special cases, for $r = 1, 2$, the convergence is said to be at least Q-superlinear, Q-superquadratic respectively. □

2.2 Newton's Method and Line Search

The optimality conditions described in the previous section serve as a general guidance for finding the local minimizers of both (NEP) and (NIP). In order to find points satisfying the first-order optimality conditions, thus be the candidates of local minimizers, Newton's method or its variations provide a powerful tool.

2.2.1 Newton's Method

Newton's method is a well-known and effective zero-finding approach, and its modified variants are used throughout this dissertation. Given a continuously differentiable function $f(x) : \mathbb{R}^n \mapsto \mathbb{R}$, its unconstrained local minimizers can be found among the solutions of $\nabla f(x) = 0$.

Given an appropriate starting point x_0, Newton's method is an iterative method that generates a sequence $\{x_k\}_{k\geq 0}$, under certain conditions, which will converge to the solutions of $\nabla f(x) = 0$ at a local quadratic rate. The Newton iteration is given by

$$\nabla^2 f(x_k)(x_{k+1} - x_k) + \nabla f(x_k) = 0. \tag{2.2}$$

A typical Newton's method is described in Algorithm 2.1.

Algorithm 2.1 Conventional Newton's Method

```
Choose $x_0$;   $k \leftarrow 0$;
while not converged do
    Evaluate $f(x_k)$, $\nabla f(x_k)$, $\nabla^2 f(x_k)$;
    Solve $\nabla^2 f(x_k) p_k = -\nabla f(x_k)$;
    Set $x_{k+1} \leftarrow x_k + p_k$;
    $k \leftarrow k + 1$;
end while
```

In the context of optimization, Newton's method may also be regarded as exploiting the *local quadratic model* of $f(x)$ at x_k, i.e.,

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

Any local minimizer of $q_k(x)$ must satisfy $\nabla q_k(x) = 0$, i.e., $\nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) = 0$, which is the same as the Newton equation (2.2). Newton's method is attractive due to its possible local quadratic convergence rate under certain conditions. Theorem 2.2.1 below gives the local convergence property of Newton's method whose proof could be found in standard optimization books.

Theorem 2.2.1 (Local convergence of Newton's method). *Let $f(x) : \mathcal{D} \subset \mathbb{R}^n \mapsto \mathbb{R}$ be twice continuously differentiable on an open convex set $\mathcal{D}$, and assume that $\nabla f(x^*) = 0$ for some $x^* \in \mathcal{D}$ with $\nabla^2 f(x^*)$ nonsingular. Then there exists an open neighborhood $\mathcal{B}$ with $x^* \in \mathcal{B}$ such that for any $x_0 \in \mathcal{B}$, the Newton iterates $\{x_k\}_{k\geq 0}$ are well-defined, remain in $\mathcal{B}$ and converge to x^* with Q-superlinear convergence. If in addition, $\nabla^2 f(x)$ is locally Lipschitz at x^*, i.e., if there exists an $L > 0$ such that $||\nabla^2 f(x) - \nabla^2 f(x^*)|| \leq L||x - x^*||$, for all $x \in \mathcal{B}$, then $\{x_k\}_{k\geq 0}$ converges to x^* with Q-quadratic convergence.* □

However, besides the rapid convergence to a stationary point, Newton's method may also diverge or there exists some p_k that is not well defined. Furthermore, Newton's method is intent on solving $\nabla f(x) = 0$, not intent on minimizing $f(x)$ since no requirements on the Hessian is imposed. So it may converge to a local maximizer or saddle point unless $f(x)$ is strictly convex on domain $\mathcal{D}$.

Moreover, in Newton's method, the starting point x_0 is required to be sufficiently close to x^*, a good guess of x_0 can be hard if there is nothing known about x^*. In this case, to obtain global convergence, a *merit function* is used to gauges the quality of x_k as an estimate of x^*. The idea of merit function will be discussed in later chapters.

In the next part, line search methods will be described that are often combined with Newton's method for minimizing the merit function.

2.2.2 Model-Based Line Search Methods

As before, the local quadratic model of $f(x)$ at x_k can be described as

$$q_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \tfrac{1}{2}(x - x_k)^T B_k(x - x_k),$$

where B_k is either $\nabla^2 f(x_k)$ or its approximation. Line search method ensures that $f(x_{k+1}) < f(x_k)$, by computing a *step* $\alpha_k > 0$ such that $f(x_k + \alpha_k p_k) < f(x_k)$. The process of finding α_k is called *line search*.

A typical structure of the model-based line search method can be described in the following Algorithm 2.2. In Algorithm 2.2, γ_c is called the *contraction factor* that is used to cut back the step α_k and η_s is called the *reduction factor* that guarantees the actual reduction in $f(x)$ will be no less than η_s times the reduction predicted by model $q_k(x)$. Step 7 to Step 9 is called *backtracking line search* along p_k.

Algorithm 2.2 A Model-based Line Search Method

```
 1: Specify constants $0 < \eta_s < 1$, and $0 < \gamma_c < 1$;
 2: $k \leftarrow 0$;
 3: while not converged do
 4:     $p_k = \text{argmin}_d \left\{ q_k(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T B_k d \right\}$;
 5:     $\alpha_k \leftarrow 1$;
 6:     $\rho_k = (f(x_k + \alpha_k p_k) - f(x_k)) \,/\, (q_k(x_k + \alpha_k p_k) - q_k(x_k))$;
 7:     while $\rho_k < \eta_s$ do
 8:         $\alpha_k \leftarrow \gamma_c \alpha_k$;
 9:         $\rho_k = (f(x_k + \alpha_k p_k) - f(x_k)) \,/\, (q_k(x_k + \alpha_k p_k) - q_k(x_k))$;
10:     end while
11:     $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
12:     $k \leftarrow k + 1$;
13: end while
```

In Algorithm 2.2, it is worthwhile pointing out that the model of reduction in $f(x)$ as in Step 9 needs not be the same as the model used to define p_k as in Step 4. When the model of reduction in $f(x)$ is linear, the sufficient decrease condition becomes

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \eta_s \alpha_k \nabla f(x_k)^T p_k$$

which is called the *Armijo condition.* An alternative to line search method will be the trust-region method, in which the search direction and step are computed in one shot within a trust region, within which the model is regarded as a "trusted" model of $f(x)$. Trust-region method will be used in Chapter 5 as an alternative to line search for minimizing the merit function.

Chapter 3

A Primal-Dual Path-Following Augmented Lagrangian Method

3.1 Introduction

In this chapter, a primal-dual path-following augmented Lagrangian method (PDAL) is proposed for solving (NEP). This is an iterative method where at each iteration, a Newton-like method is used to solve a perturbed optimality condition that defines a penalty trajectory parameterized by both the penalty parameter and the estimated Lagrange multipliers. A primal-dual augmented Lagrangian function is also defined as a merit function to guarantee global convergence. It can be shown that this method is globally convergent and under certain conditions, has a local quadratic convergence rate in the limit.

Section 3.2 provides the historical notes of methods for solving (NEP), which serves as a general background. The path-following method of Armand and Omheni [3] is based on minimizing the quadratic penalty function and is described in Section 3.3 as motivation. The remainder of this chapter describes the proposed PDAL, and its convergence results. Finally, numerical results from the CUTEst test collection are given in Chapter 6.

3.2 Background

This chapter focuses on solving the following problem, where both $f(x)$ and $c(x)$ are assumed to be twice continuously differentiable.

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

Two prominent methods, the penalty function method and the augmented Lagrangian method are described in this section. In addition, a general background of the proposed primal-dual path-following augmented Lagrangian method is presented.

3.2.1 The Penalty Function Method

The solutions of problem (NEP) can be obtained by solving a sequence of *unconstrained* problems parameterized by a scalar μ. The class of penalty function method adopts this idea by minimizing a sequence of penalty functions with the penalty parameters adjusted dynamically. Among this class, the *quadratic* penalty method is one of the oldest idea which may be traced back to a paper of Courant [7] in 1943 and a theoretical analysis by Fiacco and McCormick [8, 9].

The conventional quadratic penalty function is formally defined as

$$P_2(x; \mu) = f(x) + \frac{1}{2\mu}||c(x)||^2,$$

where $\mu > 0$ is called the *penalty parameter*, which is defined to penalize the sum of squares of constraint violations.

In the conventional quadratic penalty function method, $P_2(x; \mu)$ is minimized as an unconstrained problem for a decreasing sequence $\{\mu_k\}_{k \geq 0} \to 0^+$. Denote $\{x(\mu_k)\}_{k \geq 0}$ to be the sequence of unconstrained local minimizers of $P_2(x; \mu)$, and let x^* be an arbitrary limit point of $\{x(\mu_k)\}_{k \geq 0}$, then x^* solves (NEP). To define the Newton iteration for finding $\{x(\mu_k)\}_{k \geq 0}$, the gradient and Hessian of the quadratic penalty function $P_2(x; \mu)$ are needed, which are given by

$$\begin{aligned}\nabla P_2(x;\mu) &= g(x) + \frac{1}{\mu} J(x)^T c(x)\\ \nabla^2 P_2(x;\mu) &= \nabla^2 f(x) + \frac{1}{\mu}\sum_{i=1}^{m}(c_i(x)H_i(x)) + \frac{1}{\mu}J(x)^T J(x)\\ &= H(x,\pi) + \frac{1}{\mu}J(x)^T J(x),\end{aligned}$$

where $\pi(x) = -c(x)/\mu$ is a vector of *penalty multipliers*. At each iterate x_k, the Newton iteration is given by

$$\left(H(x,\pi) + \frac{1}{\mu}J(x)^T J(x)\right)\mathit{\Delta} x = -\left(g(x) + \frac{1}{\mu}J(x)^T c(x)\right), \tag{3.1}$$

where $H(x,\pi) = \nabla^2 f(x) - \sum_{i=1}^{m}\pi_i \nabla^2 c_i(x)$.

The conventional quadratic penalty function method has very poor numerical performance as the penalty parameter $\mu \to 0^+$. For many years, the ill-conditioning of the Hessian $\nabla^2 P_2(x;\mu) = H(x,\pi) + \frac{1}{\mu}J(x)^T J(x)$ was thought to be the cause. To see this, consider the nondegenerate case, i.e., where $J(x)$ has full row rank, the eigenvalues of $\nabla^2 P_2(x;\mu)$ when μ is sufficiently small can be characterized as:

$$\begin{aligned} m \text{ eigenvalues} &\approx \frac{1}{\mu}\lambda\left(J(x)^T J(x)\right)\\ n \text{ eigenvalues} &\approx \lambda\left(Z^T H(x,\pi) Z\right),\end{aligned} \tag{3.2}$$

where $J(x)Z = 0$, and columns of Z are orthonormal. $\lambda(H)$ denotes the eigenvalues of H here. When μ is sufficient close to 0, the conditional number $\text{cond}(\nabla^2 P_2(x;\mu)) \approx O(1/\mu)$, which is unbounded as $\mu \to 0^+$. This implies that the Newton equations associated with finding a zero of the penalty function gradient $\nabla P(x;\mu)$ become increasingly ill-conditioned as $\mu \to 0^+$.

However, this ill-conditioning is not the main reason for the poor performance of the conventional quadratic penalty function method. To see this, define the auxiliary variable $w = \frac{1}{\mu}(J(x)\mathit{\Delta} x + c(x))$, then the Newton equation $\nabla^2 P_2(x;\mu)p = -\nabla P_2(x;\mu)$ can be rewritten in the following equivalent form

$$\begin{pmatrix} H(x,\pi) & J(x)^T \\ J(x) & -\mu I \end{pmatrix} \begin{pmatrix} \Delta x \\ w \end{pmatrix} = - \begin{pmatrix} g(x) \\ c(x) \end{pmatrix},$$

which is well conditioned if problem (NEP) is well conditioned.

The two relations in (3.2) indicate that when $\mu \approx 0$, $P_2(x;\mu)$ has a small curvature in the null space of $J(x)$, denoted by null$(J(x))$, but very large curvature in the space that is orthogonal to null$(J(x))$. This means that $P_2(x;\mu)$ may vary slowly along the vectors in null$(J(x))$ but rapidly along the vectors that are orthogonal to null$(J(x))$. This implies poor convergence if $x_k \to x^*$ in the tangent space.

Another perspective of the conventional quadratic penalty function method is to consider the following *perturbed* optimality conditions

$$\begin{pmatrix} g(x) - J(x)^T y \\ c(x) + \mu y \end{pmatrix} = 0. \tag{3.3}$$

Notice that without μy, the second equation in (3.3), would be the same as the first-order optimality condition for (NEP). Thus the term μy can be seen as a *shift* to the equality constraints. As $\mu > 0$ decreases, the solution $(x(\mu), y(\mu))$ of (3.3) defines a smooth *penalty trajectory* that passes through the solution (x^*, y^*).

When μ_k is updated by μ_{k+1}, i.e., $\mu_{k+1} < \mu_k$, the first Newton step Δx_k may not always be parallel to the tangent of penalty trajectory at x_k, which makes $x_k + \Delta x_k$ generally a poor starting point for finding the next iterate x_{k+1}. To explain this further, consider any $(x(\mu), y(\mu))$ on the penalty trajectory, i.e., $(x(\mu), y(\mu))$ satisfies the following equation defined by

$$\begin{pmatrix} g(x(\mu)) - J(x(\mu))^T y(\mu) \\ c(x(\mu)) + \mu y(\mu) \end{pmatrix} = 0. \tag{3.4}$$

Differentiating both sides of equation (3.4) with respect to μ gives

$$\begin{pmatrix} H(x(\mu), y(\mu)) & -J(x(\mu))^T \\ J(x(\mu)) & \mu I \end{pmatrix} \begin{pmatrix} x'(\mu) \\ y'(\mu) \end{pmatrix} = - \begin{pmatrix} 0 \\ y(\mu) \end{pmatrix}. \tag{3.5}$$

A simple rearrangement gives

$$\left(H(x(\mu), y(\mu)) + \frac{1}{\mu} J(x(\mu))^T J(x(\mu)) \right) x'(\mu) = -\frac{1}{\mu} J(x(\mu))^T y(\mu).$$

If (x_k, y_k) is on the penalty trajectory, i.e., $x_k = x(\mu_k)$, $y_k = y(\mu_k)$, then the tangent at x_k, denoted by $x'(\mu_k)$, is given by

$$\left(H(x_k, y_k) + \frac{1}{\mu_k} J_k^T J_k \right) x'(\mu_k) = -\frac{1}{\mu_k} J_k^T y_k. \tag{3.6}$$

However, the Newton step $\mathit{\Delta} x_k$ defined in equation (3.5) is given by

$$\begin{aligned} \left(H(x_k, \pi_{k+1}) + \frac{1}{\mu_{k+1}} J_k^T J_k \right) \mathit{\Delta} x_k &= - \big(g(x_k) - J_k^T \pi_{k+1} \big) \\ &= - \big(g(x_k) - J_k^T \pi_k + J_k^T \pi_k - J_k^T \pi_{k+1} \big) \\ &= - \left(J_k^T y_k - \frac{\mu_k}{\mu_{k+1}} J_k^T y_k \right) \\ &= -J_k^T y_k \left(1 - \frac{\mu_k}{\mu_{k+1}} \right). \end{aligned} \tag{3.7}$$

The third equation holds because $\pi_k = -c_k/\mu_k = y_k$ on the penalty trajectory. Compare equation (3.6) with (3.7), since $\pi_{k+1} \neq \pi_k$, $x'(\mu_k)$ and $\mathit{\Delta} x_k$ may be quite different. This implies that along the Newton direction, $x_k + \mathit{\Delta} x_k$ might move away from the penalty trajectory and many line-search iterations are needed to drag the iterates back. Thus the Newton step $\mathit{\Delta} x_k$ is rejected and a conventional penalty function method will be inevitably inefficient.

A general scheme of the conventional quadratic penalty method has been given in Algorithm 3.1. In Step 8, the Armijo-type line search is used to guarantee the sufficient decrease condition on $P(x; \mu)$ is satisfied. The positive definite matrix E_k in Step 5 is introduced to serves as a "modification" of $\nabla^2 P_2(x_k;\mu)$ to make $\nabla^2 P_2(x_k;\mu) + E_k$ positive definite, so the Newton step p_k computed form Step 6 is guaranteed to be a *descent direction* of $P_2(x;\mu)$. In practice, E_k can be chosen to be a positive diagonal matrix, whose diagonals are increased until $\nabla^2 P_2(x_k;\mu) + E_k$ is sufficiently positive definite.

Algorithm 3.1 Conventional Quadratic Penalty Function Method

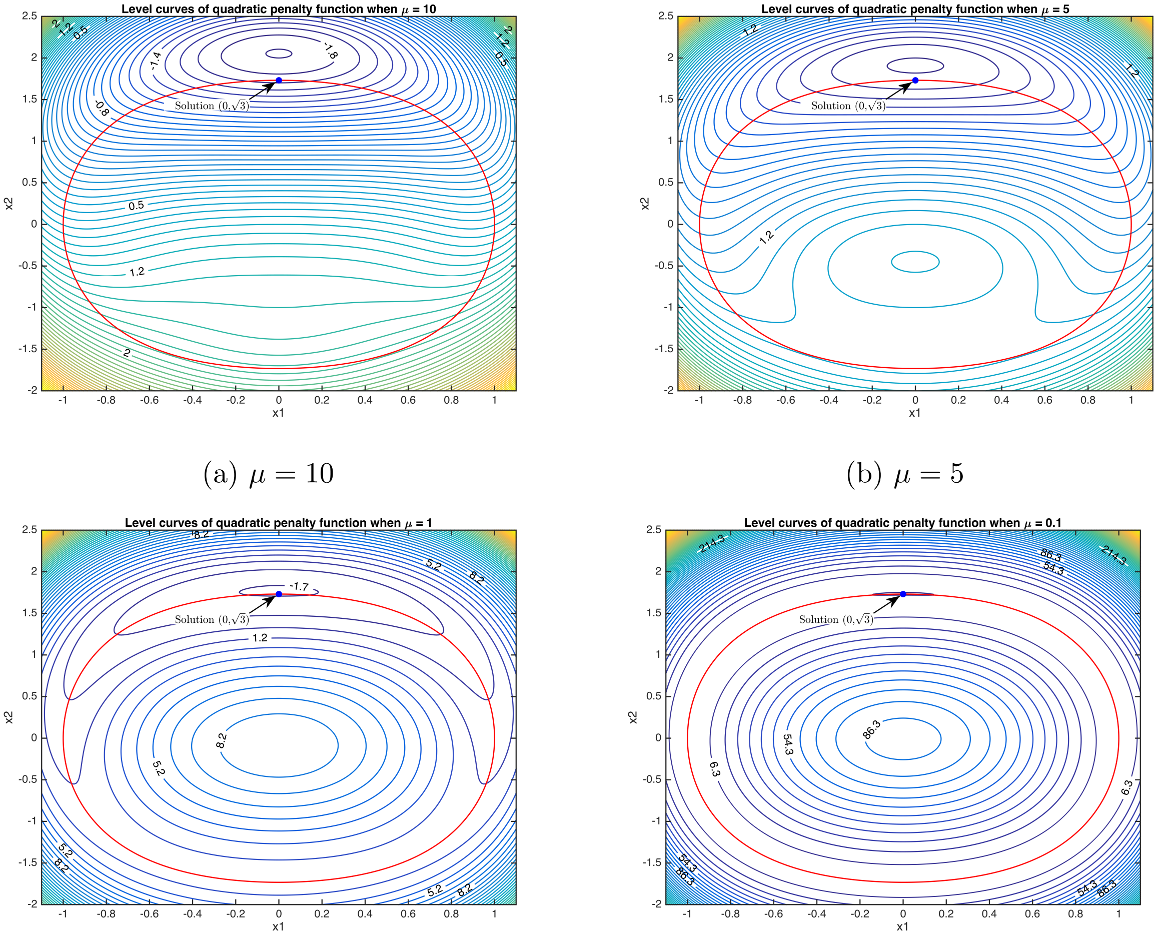
```
1: Choose constants $\eta_s, \gamma_c, \gamma, \varepsilon$ with $0 < \eta_s < \frac{1}{2}$, $0 < \gamma_c, \gamma < 1$, and $0 < \varepsilon \ll 1$;
2: Choose $x_0$, initial penalty parameter $\mu_0 > 0$, and $k \leftarrow 0$;
3: while not converged do
4:     while $||\nabla P_2(x_k;\mu)|| > \varepsilon$ do
5:         Define positive definite matrix $E_k$ such that $\nabla^2 P_2(x_k;\mu) + E_k$ is positive definite;
6:         Solve $(\nabla^2 P_2(x_k;\mu) + E_k)\, p_k = -\nabla P_2(x_k;\mu)$;                ▷ Newton step
7:         Set initial step $\alpha_k \leftarrow 1$;
8:         while $P_2(x_k + \alpha_k p_k;\mu_k) > P_2(x_k;\mu_k) + \eta_s \alpha_k \nabla P_2(x_k;\mu_k)^T p_k$ do
9:             $\alpha_k \leftarrow \gamma_c \alpha_k$;                ▷ Line search along $p_k$
10:        end while
11:        $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
12:        $k \leftarrow k + 1$;
13:    end while
14:    $\mu_{k+1} \leftarrow \gamma \mu_k$;                ▷ Decrease penalty parameter
15: end while
```

To explain the penalty trajectory intuitively, consider the following example HS7 from the Hock-Schittkowski (HS) test collection. The HS problems are an important subclass of the Constrained and Unconstrained Testing Environment (CUTEst) test collection, which is a commonly used environment for testing optimization algorithms (see Chapter 6 for more details). This example is used repeatedly throughout this chapter.

Example 3.2.1. *Consider the following nonlinear equality constrained problem HS7 in two variables.*

$$
\text{(HS7)} \qquad \begin{aligned} &\underset{x\in\mathbb{R}^2}{\text{minimize}} \quad \ln(1+x_1^2) - x_2 \\ &\text{subject to} \;\; (1+x_1^2)^2 + x_2^2 - 4 = 0. \end{aligned}
$$

The unique isolated local (and also global) minimizer of HS7 is $x^* = (0, \sqrt{3})$ with unique optimal Lagrange multipliers $y^* = -\sqrt{3}/6$, computed from $g(x^*) = J(x^*)^T y^*$. The level curves of $P(x;\mu)$ for a sequence of decreasing μ are shown in Figure 3.1.



(a) $\mu = 10$

(b) $\mu = 5$

(c) $\mu = 1$

(d) $\mu = 0.1$

Figure 3.1: Level Curves of the Conventional Quadratic Penalty Function for Different Values of μ.

In Figure 3.1, the red elliptic is the feasible region and the blue point represents the solution *of HS7. It can be observed that as $\mu \to 0^+$, the level curves of $P(x;\mu)$ gradually resemble the level curves of the constraint function and a sequence of its unconstrained local minimizers approach x^* vertically from above.* □

Theorem 3.2.1 provides the main convergence result of the conventional quadratic function method, whose proof can be referred in Fiacco and McCormick [8].

Theorem 3.2.1. *Suppose that $f(x), c(x) \in \mathcal{C}^2$, with $y_k = -c(x_k)/\mu_k$ being the penalty multipliers, such that $||\nabla P(x_k;\mu_k)|| \le \varepsilon_k$ where $\varepsilon_k \to 0$ as $k \to \infty$, and that $\{x_k\}_{k\ge 0}$ converges to x^* for which $J(x^*)$ has full rank. Then x^* satisfies the first-order optimality conditions for (NEP) (see Theorem 2.1.1) and $\{y_k\}_{k\ge 0}$ converges to the associated Lagrange multipliers y^*.* □

3.2.2 Augmented Lagrangian Method

The augmented Lagrangian method was first proposed by Hestenes [30] and Powell [36], and then developed by Rockafellar [42], Bertsekas [5] and others. Powell derived the augmented Lagrangian method as a shifted penalty function method for which the penalty parameter μ does not need to converge to zero. The price that has to be paid for keeping μ bounded away from zero is the need to update the estimated Lagrange multipliers at each iteration. The augmented Lagrangian method is known to be robust in the case where $J(x)$ does not have full rank, i.e., gradient of the constraints are linearly dependent.

The relation of the penalty function method and the augmented Lagrangian method can be understood by considering a problem in which the equality constraint $c(x) = 0$ is replaced by the *shifted* constraint $c(x) - \mu y^E = 0$, i.e.,

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) - \mu y^E = 0, \tag{PNEP}$$

where y^E is the estimated Lagrange multipliers. Then the penalty function for solving problem (PNEP) is given by

$$\begin{aligned}\widehat{P}(x;y^{E},\mu) &= f(x)+\frac{1}{2\mu}||c(x)-\mu y^{E}||^2\\ &= f(x)+\frac{1}{2\mu}\left(||c(x)||^2-2\mu c(x)^T y^{E}+\mu^2||y^{E}||^2\right).\end{aligned}$$

For a fixed pair (y^{E},μ), minimizing $\widehat{P}(x;y^{E},\mu)$ is equivalent to minimizing the *augmented Lagrangian function* defined as

$$L_{A}(x;y^{E},\mu) = f(x)-c(x)^T y^{E}+\frac{1}{2\mu}||c(x)||^2.$$

The main theoretical basis of the conventional augmented Lagrangian method is stated in Theorem 3.2.2, which also indicates that μy^* would be the optimal shift, where y^* is the optimal Lagrange multipliers of (NEP).

Theorem 3.2.2. *Suppose x^* satisfies the second-order sufficient conditions for a minimizer of (NEP) (see Theorem 2.1.3). Let y^* be the corresponding optimal Lagrange multipliers at x^*. Then there must exist a finite $\bar{\mu}>0$, for every $0<\mu<\bar{\mu}$, a solution x^* of (NEP) is an isolated unconstrained minimizer of $L_{A}(x;y^*,\mu)$.* □

Proof. By definition, $L_{A}(x;y^*,\mu)=f(x)-c(x)^Ty^*+\frac{1}{2\mu}||c(x)||^2$, the gradient and Hessian of L_{A} are given by

$$\begin{aligned}\nabla L_{A}(x;y^*,\mu) &= g(x)-J(x)^T(y^*-\frac{1}{\mu}c(x))\\ \nabla^2 L_{A}(x;y^*,\mu) &= H(x,y^*-\frac{1}{\mu}c(x))+\frac{1}{\mu}J(x)^TJ(x).\end{aligned} \tag{3.8}$$

The first-order optimality condition implies that if x^* is a solution of (NEP), then $c(x^*)=0$ and $g(x^*)-J(x^*)^Ty^*=0$, which further implies that $\nabla L_{A}(x^*;y^*,\mu)=0$ and $\nabla^2 L_{A}(x^*;y^*,\mu)=H(x^*,y^*)+\frac{1}{\mu}J(x^*)^TJ(x^*)$. The second-order sufficient conditions (see Theorem 2.1.3) imply that the reduced Hessian $Z^TH(x^*,y^*)Z$ must be positive definite, where the columns of Z form the basis of null$(J(x^*))$. According to Debreu's Lemma (see Lemma 1.4.2), $\nabla^2 L_{A}(x;y^*,\mu)=H(x^*,y^*)+\frac{1}{\mu}J(x^*)^TJ(x^*)$ must be positive definite for all

$0 < \mu < \bar{\mu}$ with some $\bar{\mu} > 0$. Thus x^* must be an isolated local minimizer of (NEP). □

Based on Theorem 3.2.2, Hestenes and Powell proposed that x^* be found by minimizing $L_A(x; y^E, \mu)$ for a sequence of values of (y^E, μ). For a given pair (y^E, μ), $L_A(x; y^E, \mu)$ may be minimized by either a line-search or trust-region method for unconstrained optimization. Once the minimization is complete, (y^E, μ) are updated to encourage convergence. Ideally, the update should allow $y^E \to y^*$ in the limit, with μ bounded below.

From the proof of Theorem 3.2.2, we know that when minimizing $L_A(x; y^E, \mu)$, the associated Newton equation can be expressed by

$$\left(H(x, \pi(x)) + \frac{1}{\mu} J(x)^T J(x) \right) p = -\left(g(x) - J(x)^T \pi(x) \right). \tag{3.9}$$

A comparison of (3.1) with (3.9) indicates that although they share the "same" structure, the multiplier estimate $\pi(x)$ is different, with $\pi(x) = -c(x)/\mu$ in the quadratic penalty function method and $\pi(x) = y^E - c(x)/\mu$ in the augmented Lagrangian method. The primal-dual form of the Newton equations (3.9) may be written as

$$\begin{pmatrix} H(x, \pi(x)) & J(x)^T \\ J(x) & -\mu I_m \end{pmatrix} \begin{pmatrix} \mathit{\Delta} x \\ -\mathit{\Delta} y \end{pmatrix} = -\begin{pmatrix} g(x) - J(x)^T y \\ c(x) + \mu(y - y^E) \end{pmatrix}, \tag{3.10}$$

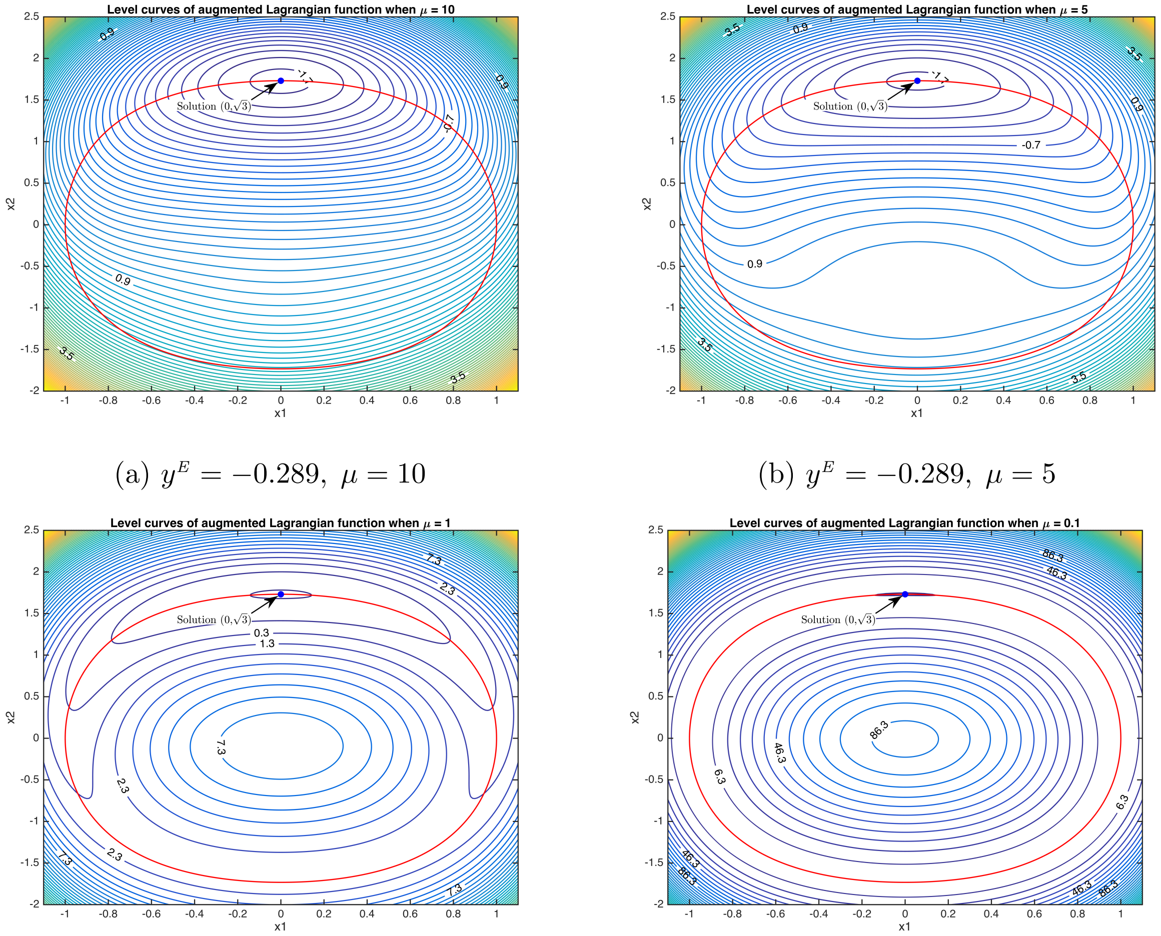
where $\mathit{\Delta} y = -\frac{1}{\mu}\left(J(x)\mathit{\Delta} x + (c(x) + \mu(y - y^E))\right)$. The diagonal matrix $-\mu I_m$ in the $(2,2)$-block of the KKT matrix serves as a *regularization* of the KKT system (3.10).

A complete description of the augmented Lagrangian method is given in Algorithm 3.2. The modification of $\nabla^2 L_A(x_k; y_k^E, \mu_k)$ by adding some positive definite E_k is used to guarantee that p_k is a descent direction for $L_A(x; y^E, \mu)$. In practice, E_k can be computed "on the fly" while computing the symmetric indefinite factorization of K_μ (See Forsgren and Gill [10]). Given the decreasing sequence $\{\beta_k\}_{k \geq 0}$, if $||c(x_{k+1})|| > \beta_k$, then μ_k is reduced to penalize the constraint violations and enforce feasibility (see Step 13). For efficiency, the multiplier estimate y^E is frequently updated in Step 17 if the constraint violations are less than the current tolerance ε_k.

Algorithm 3.2 Augmented Lagrangian Method for (NEP)

```
Choose constants $\eta_s, \gamma_c, \gamma, \alpha, \varepsilon$ with $0 < \eta_s < \frac{1}{2}$, $0 < \gamma_c, \gamma, \alpha < 1$, and $0 < \varepsilon \ll 1$;
Choose $x_0, y_0, y_0^E, \mu_0 > 0$, and $\beta_0 = \alpha||c(x_0)||$;
Initial iteration $k \leftarrow 0$;
while not converged do
    Define positive definite matrix $E_k$ such that $\nabla^2 L_A(x_k; y_k^E, \mu_k) + E_k$ is positive definite;
    Solve $\left(\nabla^2 L_A(x_k; y_k^E, \mu_k) + E_k\right) p_k = -\nabla L_A(x_k; y_k^E, \mu_k)$;    ▷ Newton step
    Set initial step $\alpha_k \leftarrow 1$;
    while $L_A(x_k + \alpha_k p_k; y_k^E, \mu_k) > L_A(x_k; y_k^E, \mu_k) + \eta_s \alpha_k \nabla L_A(x_k; y_k^E, \mu_k)^T p_k$ do
        $\alpha_k \leftarrow \gamma_c \alpha_k$;    ▷ Line search along $p_k$
    end while
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
    if $||\nabla L_A(x_{k+1}; y_k^E, \mu_k)|| \le \varepsilon_k$ then
        if $||c(x_{k+1})|| > \beta_k$ then
            $\mu_{k+1} \leftarrow \gamma \mu_k$;    ▷ Decrease $\mu$ to enforce feasibility
            $\beta_{k+1} \leftarrow \alpha||c(x_{k+1})||$, $\varepsilon_{k+1} \leftarrow \frac{1}{2}\varepsilon_k$;
        else
            $y_{k+1}^E \leftarrow y_k^E$, $\mu_{k+1} \leftarrow \mu_k$;    ▷ Update multipliers estimate
            $\beta_{k+1} \leftarrow \beta_k$, $\varepsilon_{k+1} \leftarrow \varepsilon_k$;
        end if
    end if
    $k \leftarrow k + 1$;
end while
```

Example 3.2.2. *Consider applying the augmented Lagrangian method to solve problem HS7* *again. Figure 3.2 shows the level curves of $L_A(x; y^E, \mu)$ for $y^E = -0.289$, which is the approximation of the optimal multipliers $y^* = -\sqrt{3}/6$, and a decreasing sequence of μ.*



(a) $y^E = -0.289,\ \mu = 10$

(b) $y^E = -0.289,\ \mu = 5$

(c) $y^E = -0.289,\ \mu = 1$

(d) $y^E = -0.289,\ \mu = 0.1$

Figure 3.2: Level Curves of augmented Lagrangian Function for Different Values of μ.

In Figure 3.2, the colored level curves of the augmented Lagrangian function $L_A(x; y^E, \mu)$ are given for a decreasing sequence of μ. The red elliptic is the feasible region and the blue point represents the solution of HS7. It can be shown that when μ is sufficiently small, then x^* is an unconstrained local minimizer of $L_A(x; y^*, \mu)$. This implies that x^* is "contained" in the center of level curves of $L_A(x; y^E, \mu)$ and there is no need for reducing μ to zero. The penalty trajectory will pass through x^* as μ decreases to some sufficient small number. □

3.3 Motivation of The Proposed Algorithm

This section describes a primal-dual path-following method for (NEP), proposed by Armand et al. [2], which is a Newton-like method applied to a sequence of perturbed optimality systems that follow naturally from the quadratic penalty approach. Their algorithm has a “two-level”structure in which the outer iteration involves solving a perturbed KKT linear system and monitoring the residual. If the residual is not sufficiently small, a quadratic penalty function is minimized in the inner iteration to obtain global convergence. It has been shown that whenever the primal-dual pair (x_k, y_k) converges to a regular solution (x^*, y^*), if μ_k converges to 0 with a superlinear rate of convergence, then the algorithm reduces asymptotically to Newton iterations and the inner iteration is no longer needed.

The first-order optimality condition for minimizing $P_2(x; \mu)$ is given by

$$g(x) + \frac{1}{\mu} J(x)^T c(x) = 0. \tag{3.11}$$

If $y = -c(x)/\mu$ is used as an auxiliary variable, then (3.11) is equivalent to the perturbed optimality condition for (NEP):

$$F(x, y; \mu) = \begin{pmatrix} g(x) - J(x)^T y \\ c(x) + \mu y \end{pmatrix} = 0. \tag{3.12}$$

Define the primal-dual pair $v(\mu) = (x(\mu), y(\mu))$, then $F(x, y; \mu) = 0$ implicitly defines a trajectory $\mu \to v(\mu)$ which passes through the solution (x^*, y^*) as $\mu \to 0$. The main idea is to define a sequence of primal-dual iterates (x_k, y_k) that closely path-follows this trajectory. Specifically, in the outer iteration, the idea is to apply a Newton-like method on $F(x, y; \mu_k) = 0$ for some decreasing sequence $\mu_k > 0$. A steadily decreasing sequence of $\varepsilon_k > 0$ is defined so that if the trial step $||F(x_k^+, y_k^+; \mu_k)|| \leq \varepsilon_k$, then (x_k^+, y_k^+) is accepted as the next iterate. Otherwise the inner iteration is called where a penalty type merit function that measures the distance of the current iterate to the trajectory is minimized by applying a backtracking line search to obtain global convergence.

The merit function $\psi_\mu(v)$ used in the inner iteration is the primal-dual penalty function proposed by Forsgren and Gill [11]. This function is the conventional quadratic penalty function plus a term that measures the distance of the current iterate to the path-following trajectory.

$$\psi_\mu(v) = f(x) + \frac{1}{2\mu}||c(x)||^2 + \frac{\nu}{2\mu}||c(x) + \mu y||^2, \tag{3.13}$$

where $\nu > 0$ is a scaling parameter to balance the penalty of the constraint violations and the distance to the path-following trajectory. v is the primal-dual pair defined as $v = (x, y)$. The gradient and Hessian of $\psi_\mu(v)$ are given by

$$\nabla\psi_\mu(v) = \begin{pmatrix} g(x) + J(x)^T(\frac{1}{\mu}c(x) + \frac{\nu}{\mu}(c(x) + \mu y)) \\ \nu(c(x) + \mu y) \end{pmatrix}$$

$$\nabla^2\psi_\mu(v) = \begin{pmatrix} H(x, -(\frac{1}{\mu}c(x) + \frac{\nu}{\mu}(c(x) + \mu y))) & \nu J^T \\ \nu J & \nu\mu I_m \end{pmatrix}.$$

It is immediate that every primal-dual pair $v = (x, y)$ on the trajectory defined by the path-following equation (3.12) is a stationary point of $\psi_\mu(v)$. Furthermore, if the second order sufficient conditions of (NEP) (see Theorem 2.1.3) hold at the primal-dual pair $v^* = (x^*, y^*)$, then $\nabla^2\psi_\mu(v^*)$ must be positive definite, which guarantees that the local minimum could be found.

In a subsequent paper [3], Armand and Omheni extended this method to a primal-dual augmented Lagrangian method for solving the equality constrained minimization problem. This is also a Newton-type method applied to the perturbed optimality conditions but it based on the properties of the augmented Lagrangian function.

The next section describes an alternative perturbed optimality condition that may also be regarded as defining an implicit penalty trajectory. A primal-dual augmented Lagrangian merit function is minimized to keep the iterates within a certain neighborhood of the trajectory.

3.4 Description of the Proposed Algorithm

In this section, a detailed description of the proposed primal-dual path-following augmented Lagrangian method (PDAL) is given. This method shares a similar "two-level" structure containing both the outer and inner iterations. The idea is to define a trajectory by a perturbed *first-order* optimality condition and then define a merit function with the potential of giving limit point that satisfies the *second-order* necessary optimality conditions. PDAL is also in part motivated by the Forsgren and Gill primal-dual interior method (See [10]), but is intended to solve only equality constrained problems from a *path-following perspective.*

In the outer iteration of PDAL, a trial iterate (x_k^+, y_k^+) is obtained by applying the modified Newton's method for finding the approximate solution of $F(x, y; y^E, \mu) = 0$, which is defined by

$$F(x, y; y^E, \mu) = \begin{pmatrix} g(x) - J(x)^T y \\ c(x) + \mu(y - y^E) \end{pmatrix}, \tag{3.14}$$

where y^E is the current Lagrange multipliers estimate and μ is the penalty parameter. If (x_k^+, y_k^+) reduces $||F(x, y; y^E, \mu)||$ sufficiently, then it is accepted as a new iterate. Otherwise, the inner iteration is called where an augmented Lagrangian type merit function $\mathcal{M}(x, y; y_k^E, \mu)$ defined in (3.17) is minimized to restrict next iterate back to the neighborhood of the path-following trajectory.

In fact, the path-following equation (3.14) can be obtained by considering the gradient of the conventional augmented Lagrangian function $L_A(x; y^E, \mu) = f(x) - c(x)^T y^E + \frac{1}{2\mu}||c(x)||^2$, which is given by

$$\nabla L_A(x; y^E, \mu) = g(x) - J(x)^T \left(y^E - \frac{1}{\mu} c(x) \right).$$

If $y = y^E - c(x)/\mu$, then $\nabla L_A(x; y^E, \mu) = 0$ is equivalent to $F(x, y; y^E, \mu) = 0$, which is exactly (3.14). So the idea is to find zeros of $F(x, y; y^E, \mu) = 0$ while updating parameters μ and y^E in a manner that guarantees strong convergence properties.

3.4.1 Description of The Outer Iteration

In the outer iteration of PDAL, a Newton-like method is used for finding the zeros of $F(x, y; y^E, \mu) = 0$. At the current iterate (x_k, y_k), the linearization of $F(x, y; y^E, \mu)$ at (x_k, y_k, μ_k) with respect to (x, y, μ) is given by:

$$\begin{pmatrix} H(x_k, y_k) & -J(x_k)^T \\ J(x_k) & \mu_k I_m \end{pmatrix} \begin{pmatrix} x_k^+ - x_k \\ y_k^+ - y_k \end{pmatrix} + \begin{pmatrix} 0 \\ y_k - y_k^E \end{pmatrix} (\mu_k^+ - \mu_k) = - \begin{pmatrix} g(x_k) - J(x_k)^T y_k \\ c(x_k) + \mu_k (y_k - y_k^E) \end{pmatrix},$$

where $H(x_k, y_k)$ is the Hessian of Lagrangian at (x_k, y_k), and (x_k^+, y_k^+) is the next trial iterate.

An equivalent but *symmetric* linear system can be rewritten as

$$\begin{pmatrix} H(x_k, y_k) & J(x_k)^T \\ J(x_k) & -\mu_k I_m \end{pmatrix} \begin{pmatrix} x_k^+ - x_k \\ y_k - y_k^+ \end{pmatrix} + \begin{pmatrix} 0 \\ y_k - y_k^E \end{pmatrix} (\mu_k^+ - \mu_k) = - \begin{pmatrix} g(x_k) - J(x_k)^T y_k \\ c(x_k) + \mu_k (y_k - y_k^E) \end{pmatrix}.$$

A simple rearrangement gives

$$\begin{pmatrix} H(x_k, y_k) & J(x_k)^T \\ J(x_k) & -\mu_k I_m \end{pmatrix} \begin{pmatrix} x_k^+ - x_k \\ y_k - y_k^+ \end{pmatrix} = - \begin{pmatrix} g(x_k) - J(x_k)^T y_k \\ c(x_k) + \mu_k^+ (y_k - y_k^E) \end{pmatrix}. \tag{3.15}$$

If this KKT matrix is denoted by K_k, then (3.15) can be written as

$$K_k \begin{pmatrix} x_k^+ - x_k \\ y_k - y_k^+ \end{pmatrix} = -F(x_k, y_k; y^E, \mu_k^+). \tag{3.16}$$

Note that the $(2, 2)$-block of K_k is $-\mu_k I_m$ while an updated value of μ_k^+ $(\mu_k^+ < \mu_k)$ is used on the right-hand side of (3.16).

If the second-order sufficient conditions hold at (x, y) (see Theorem 2.1.3), then the reduced Hessian $Z(x)^T H(x, y) Z(x)$ must be positive definite. According to Debreu's Lemma (see Lemma 1.4.2), there must exist a constant $\bar{\mu} > 0$, such that $H(x, y) + \frac{1}{\mu} J(x)^T J(x)$ is positive definite whenever $0 < \mu < \bar{\mu}$. The update rule of μ_k is based on this observation. Specifically, if $H(x_k, y_k) + \frac{1}{\mu_k} J_k^T J_k$ is not sufficiently positive definite, then the trial $\mu_k^+ = \min \left\{ \mu_k^{1+\sigma}, a\mu_k, \frac{1}{k} \right\}$, where σ, $a \in (0, 1)$. This strategy is the same as the one used by Armand et al. [2, 3], and it can be observed numerically that as $\mu_k \to 0^+$, the linear decrease of the form $a\mu_k$ is more acceptable.

Another important issue is that if K_k has more than m negative eigenvalues, then the $(1,1)$-block need to be modified to ensure that $\mathrm{In}(K_k) = (n, m, 0)$ (and therefore that the reduced Hessian is positive definite). There are many techniques of modifying matrices of the form K_k. One example is to add a positive semidefinite matrix $\eta_k I_n$ to the $(1,1)$-block of H_k and increase η_k whenever K_k has more than m negative eigenvalues. This technique is applied in Algorithm 3.3. Other techniques include the use of the modified symmetric indefinite factorization of K_k (see, Forsgren, Gill and Murray [14], Gill and Wong [26, 27]).

Once the modified KKT system (3.16) has been solved and the next trial iterate (x_k^+, y_k^+) obtained, if $||F(x_k^+, y_k^+; y_k^E, \mu_k^+)||_\infty \le \rho_s ||F(x_k, y_k; y_k^E, \mu_k)||_\infty$ for some $0 < \rho_s < 1$, then $(x_k^+, y_k^+, \mu_k^+) \to (x_{k+1}, y_{k+1}, \mu_{k+1})$ and the Lagrange multiplier estimate is updated by $y_k^+ \to y_{k+1}^E$. Otherwise the inner iteration is used to reduce $||F(x, y; y^E, \mu)||_\infty$ by minimizing the merit function using a line-search method.

Note also that the maximum number of inner iteration is limited for efficiency. It is not worthwhile using too many inner iterations to force $||F(x_k^+, y_k^+; y_k^E, \mu_k^+)||_\infty \le \varepsilon_k$, especially in the early iterations when v_k is far away from v^*. Here $\{\varepsilon_k\}$ is a positive sequence decreasing to zero. For example ε_k can be chosen as $\varepsilon_k = \rho_s ||F(x_k, y_k; y_k^E, \mu_k)||_\infty$ for some $\rho_s \in (0, 1)$. A different strategy for the choice of ε_k was adopted by Armand [2] where a relaxation parameter $\varsigma_k > 0$ is introduced such that $\varepsilon_k = \rho_s ||F(x_k, y_k; y_k^E, \mu_k)|| + \varsigma_k$. In this case, there are no restrictions on the maximum number of inner iterations.

A detailed description of the outer iteration of PDAL is given in the following Algorithm 3.3.

Algorithm 3.3 Primal-Dual Path-Following Augmented Lagrangian Method (Outer)

```
Input: (prob);                                            ▷ Input in CUTEst format
Output: Solution $(x^*, y^*, \text{exit})$, exit indicates the exit code;
Choose $x_0$, initial penalty parameter $\mu_0$, and $k \leftarrow 0$;
Choose constants $0 < a_s, \sigma_s, \rho_s, \gamma_c < 1$, $0 < \eta_s < \frac{1}{2}$, $0 < \varepsilon_s \ll 0$, MaxItn, and MaxIItn;
while $||F(x_k, y_k; y_k^E, \mu_k)||_\infty \geq \varepsilon_s$ and $k \leq$ MaxItn do
    if $H(x_k, y_k) + \frac{1}{\mu_k} J_k^T J_k \not\succ 0$ then    ▷ Update penalty parameter
        Choose $\mu_k^+$, s.t. min $\left\{\mu_k^{1+\sigma_s}, a_s\mu_k, \frac{1}{k}\right\} \leq \mu_k^+ \leq$ min $\left\{a_s\mu_k, \frac{1}{k}\right\}$;
        Choose a symmetric $H_k \succ 0$, s.t. $\text{In}(K_k) = \text{In}\begin{pmatrix} H_k & J_k^T \\ J_k & -\mu_k I \end{pmatrix} = (n, m, 0)$;
    else
        $\mu_k^+ \leftarrow \mu_k$;
    end if
    Compute $(x_k^+, y_k^+)$ by solving $K_k(x_k^+ - x_k, y_k - y_k^+) = -F(x_k, y_k; y_k^E, \mu_k^+)$;
    Set $\varepsilon_k = \rho_s||F(x_k, y_k; y_k^E, \mu_k)||_\infty$, IItn $\leftarrow 0$;    ▷ Stopping criteria for inner iteration
    if $||F(x_k^+, y_k^+; y_k^E, \mu_k^+)||_\infty \leq \varepsilon_k$ then
        $(x_{k+1}, y_{k+1}, y_{k+1}^E, \mu_{k+1}) \leftarrow (x_k^+, y_k^+, y_k^+, \mu_k^+)$;
    else                                                  ▷ Inner iteration
        while $||F(x_{k+1}, y_{k+1}; y_k^E, \mu_k^+)||_\infty > \varepsilon_k$ and IItn $\leq$ MaxIItn do
            Apply inner iterations to minimize merit function $M(x, y; y_k^E, \mu)$ in (3.17);
        end while
        $\mu_{k+1} \leftarrow \mu_k^+$,   $y_{k+1}^E \leftarrow y_{k+1}$;
    end if
    $k \leftarrow k + 1$;
end while
```

3.4.2 Description of the Inner Iteration

If $||F(x_k^+, y_k^+; y_k^E, \mu_k^+)||_\infty > \varepsilon_k$, then an inner iteration is called and (x_k^+, y_k^+) is taken to be the starting point for minimizing the merit function $\mathcal{M}(x, y; y_k^E, \mu)$ of (3.17). The penalty parameter is allowed to increase sometimes for the benefit of minimizing $\mathcal{M}(x, y; y_k^E, \mu)$ while keeping $\mathcal{M}(x, y; y_k^E, \mu)$ non-increasing.

As above, a merit function needs to be defined whose local minimizer lies on the trajectory defined by $F(x, y; y_k^E, \mu) = 0$. It is immediate that the function $\mathcal{M}(x, y; y_k^E, \mu)$ such that

$$\mathcal{M}(x, y; y_k^E, \mu) = f(x) - c(x)^T y_k^E + \frac{1}{2\mu}||c(x)||^2 + \frac{1}{2\mu}||c(x) + \mu(y - y_k^E)||^2 \tag{3.17}$$

has this property. This function is a special case of the so-called generalized primal-dual augmented Lagrangian function proposed by Robinson [41], Gill and Robinson [25], which is defined as

$$\mathcal{M}^\nu(x, y; y_k^E, \mu) = f(x) - c(x)^T y_k^E + \frac{1}{2\mu}||c(x)||^2 + \frac{\nu}{2\mu}||c(x) + \mu(y - y_k^E)||^2,$$

where $\nu > 0$ is a fixed scalar that balances the augmented Lagrangian function and the primal-dual term. The function $\mathcal{M}^\nu(x, y; y_k^E, \mu)$ is equivalent to the Forsgren-Gill primal-dual quadratic penalty function (see Forsgren and Gill [10]) defined in terms of the shifted constraints $c(x) - \mu y^E = 0$. For simplicity, $\nu = 1$ is chosen in the merit function $\mathcal{M}(x, y; y_k^E, \mu)$ without sacrificing strong convergence results.

To simplify the notation, as y_k^E is fixed during the inner iteration, let $\mathcal{M}(x, y; y_k^E, \mu)$ be denoted by $\mathcal{M}_\mu(x, y)$. Similarly, define the first-order Lagrange multiplier estimate $\pi(x) = y_k^E - c(x)/\mu$, and a typical primal-dual pair of inner iterates as (x_i, y_i). Here, a Armijo line-search method is applied for minimizing $M_\mu(x, y)$. In contrast, a trust-region method is applied to minimize the shifted penalty-barrier merit function in Chapter 5.

First, the gradient and Hessian of the merit function $\mathcal{M}_\mu(x, y)$ need to be defined, which are given by

$$\nabla \mathcal{M}_\mu(x,y) = \begin{pmatrix} g(x) - J(x)^T(\pi + (\pi - y)) \\ \mu(y - \pi) \end{pmatrix}$$

$$\nabla^2 \mathcal{M}_\mu(x,y) = \begin{pmatrix} H(x, 2\pi - y) + \frac{2}{\mu} J(x)^T J(x) & J(x)^T \\ J(x) & \mu I \end{pmatrix}.$$

Lemma 3.4.1 below states that by minimizing $\mathcal{M}_\mu(x,y)$ while maintaining certain second-order requirements, it is possible to restrict the iterates within a certain neighborhood of the path-following trajectory defined by $F(x, y; y^E_k, \mu) = 0$.

Lemma 3.4.1. *For any fixed y^E_k and $\mu > 0$, (x_i, y_i) satisfies $F(x_i, y_i; y^E_k, \mu) = 0$ and $H(x_i, y_i) + \frac{1}{\mu} J(x_i)^T J(x_i)$ is positive definite if and only if (x_i, y_i) is an isolated local minimizer of the merit function $\mathcal{M}_\mu(x,y)$.* □

Proof. The path-following equation $F(x, y; y^E_k, \mu) = 0$ implies that $g(x) - J(x)^T y = 0$ and $c(x) + \mu(y - y^E_k) = 0$. Thus if (x_i, y_i) satisfies $F(x_i, y_i; y^E_k, \mu) = 0$, then $y_i = y^E_k - c(x_i)/\mu = \pi(x_i)$, i.e., $y_i = \pi_i$. According to the second equation and together with the relation $g(x_i) - J(x_i)^T y_i = 0$, $F(x_i, y_i; y^E_k, \mu) = 0$ is equivalent to $\nabla \mathcal{M}_\mu(x_i, y_i) = 0$, i.e., (x_i, y_i) is a stationary point of $\mathcal{M}_\mu(x,y)$.

The Hessian $\nabla^2 \mathcal{M}_\mu(x,y)$ at any point (x_i, y_i) on the path-following trajectory is given by

$$\nabla^2 \mathcal{M}_\mu(x_i,y_i) = \begin{pmatrix} H(x_i, y_i) + \frac{2}{\mu} J(x_i)^T J(x_i) & J(x_i)^T \\ J(x_i) & \mu I \end{pmatrix}. \tag{3.18}$$

For any (x_i, y_i) satisfying $F(x_i, y_i; y^E_k, \mu) = 0$, if it further satisfies that $H(x_i, y_i) + \frac{1}{\mu} J(x_i)^T J(x_i)$ is positive definite, then from the inertia relation

$$\mathrm{In}\left(\nabla^2 \mathcal{M}_\mu(x_i,y_i)\right) = (m, 0, 0) + \mathrm{In}\left(H(x_i, y_i) + \frac{1}{\mu} J(x_i)^T J(x_i)\right),$$

which is obtained by observing that $H(x_i, y_i) + \frac{1}{\mu} J(x_i)^T J(x_i)$ is the Schur complement of μI in equation (3.18). Thus $\nabla^2 \mathcal{M}_\mu(x_i, y_i)$ is positive definite if and only if $H(x_i, y_i) + \frac{1}{\mu} J(x_i)^T J(x_i)$ is positive definite. As a result, if $H(x_i, y_i) + \frac{1}{\mu} J(x_i)^T J(x_i)$ is positive definite, then (x_i, y_i) must be an isolated local minimizer of $\mathcal{M}_\mu(x, y)$. □

It should be pointed out that Step 8 of Algorithm 3.3 and Step 5 of Algorithm 3.4 are used to force $H(x_k, y_k) + \frac{1}{\mu} J(x_k)^T J(x_k)$ to be positive definite for every iterate (x_k, y_k), which guarantees that a local *minimizer* of (NEP) is obtained in the limit. According to Lemma 3.4.1, it is reasonable to minimize $\mathcal{M}_\mu(x, y)$ in inner iterations to restrict the iterates towards a gradually shrinking neighborhood of the path-following trajectory.

One challenge in inner iterations is that it becomes more difficult to minimize the merit function $\mathcal{M}_\mu(x, y)$ as $\mu \to 0^+$. The technique described in Step 2 - Step 4 of Algorithm 3.4 is introduced with this goal in mind, where μ is allowed to increase sometimes for numerical benefits. Specifically, if $||y_i - y_k^E|| \neq 0$, then define

$$\widehat{\mu_i} = \frac{||c(x_i)||_2}{||y_i - y_k^E||_2} = \underset{\mu}{\text{argmin}} \;\; \frac{1}{2\mu} ||c(x_i) + \mu(y_i - y_k^E)||^2$$

then

$$\mu_i = \begin{cases} \max\left\{\widehat{\mu_i}, \mu_{i-1}\right\}, & \text{if} \quad \widehat{\mu_i} \leq \frac{1}{k} \\ \mu_{i-1}, & \text{otherwise,} \end{cases}$$

which is similar to the strategy used by Armand et al. [2].

From the definition of $M(x, y; y_k^E, \mu)$, it is clear that increasing μ will decrease the penalty term $\frac{1}{\mu}||c(x)||^2$. So the choice of $\widehat{\mu_i}$ satisfies $\mathcal{M}_{\widehat{\mu_i}}(x_i, y_i) \leq \mathcal{M}_{\mu_i}(x_i, y_i)$, i.e., increasing the penalty parameter will not lead to an increase in the merit function while some numerical benefits could be obtained.

It can also be shown that in the inner iteration, the Newton equations for minimizing $\mathcal{M}_\mu(x, y)$ have exactly the same structure as the Newton equations for finding the zeros of $F(x, y; y_k^E, \mu) = 0$ in the outer iteration. To see this, the Hessian approximation B_i of

$\nabla^2 \mathcal{M}_\mu(x_i, y_i)$ can be defined by equating $\pi_i = y_i$ in the $(1,1)$-block, which gives

$$B_i = \begin{pmatrix} H(x_i, y_i) + \frac{2}{\mu} J(x_i)^T J(x_i) & J(x_i)^T \\ J(x_i) & \mu_i I_m \end{pmatrix}.$$

It follows that the Newton equation for finding $\nabla \mathcal{M}_\mu(x, y) = 0$ is given by

$$\begin{pmatrix} H(x_i, y_i) + \frac{2}{\mu} J(x_i)^T J(x_i) & J(x_i)^T \\ J(x_i) & \mu_i I_m \end{pmatrix} \begin{pmatrix} \mathit{\Delta} x_i \\ \mathit{\Delta} y_i \end{pmatrix} = - \begin{pmatrix} g(x_i) - J(x_i)^T(\pi_i + (\pi_i - y_i)) \\ \mu(y_i - \pi_i) \end{pmatrix}. \tag{3.19}$$

Let L be the nonsingular matrix

$$L = \begin{pmatrix} I & -\frac{2}{\mu_i} J(x_i)^T \\ 0 & I \end{pmatrix}.$$

Premultiplying both sides of equation (3.19) by L gives

$$\begin{pmatrix} H(x_i, y_i) & -J(x_i)^T \\ J(x_i) & \mu_i I_m \end{pmatrix} \begin{pmatrix} \mathit{\Delta} x_i \\ \mathit{\Delta} y_i \end{pmatrix} = - \begin{pmatrix} g(x_i) - J(x_i)^T y_i \\ \mu(y_i - \pi_i) \end{pmatrix}, \tag{3.20}$$

which is equivalent to the *symmetric* system

$$\begin{pmatrix} H(x_i, y_i) & J(x_i)^T \\ J(x_i) & -\mu_i I_m \end{pmatrix} \begin{pmatrix} \mathit{\Delta} x_i \\ -\mathit{\Delta} y_i \end{pmatrix} = - \begin{pmatrix} g(x_i) - J(x_i)^T y_i \\ \mu(y_i - \pi_i) \end{pmatrix}. \tag{3.21}$$

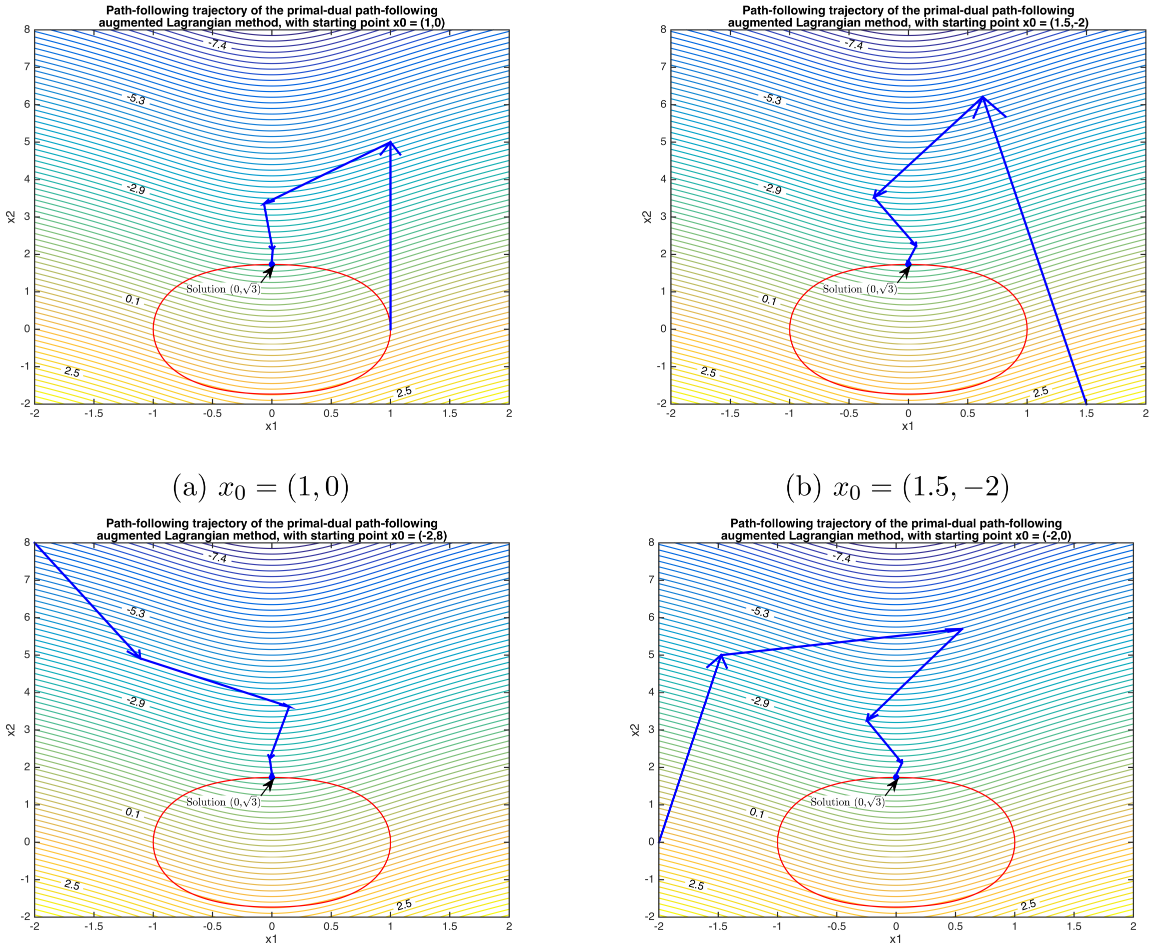
A comparison of equations (3.15) and (3.21) shows that the KKT matrices have exactly the same structure while they are different only on the right hand side. If the KKT matrix of (3.21) has more than m negative eigenvalues, its inertia must be modified using the same technique used to modify the KKT matrix in the outer iteration (see Section 3.4.1). Finally, an Armijo line search is used if the unit step fails to give a sufficient decrease of $\mathcal{M}_\mu(x, y)$. Theorem 3.5.1 states that the search direction d_i is guaranteed to be a descent direction for $\mathcal{M}_\mu(x, y)$. A complete description of inner iteration is given in Algorithm 3.4 below.

Algorithm 3.4 A Primal-Dual Path-Following Augmented Lagrangian Method (Inner)

```
while $||F(x_i, y_i; \mu_k^+, y_k^E)||_\infty \geq \varepsilon_k$ and IItn $\leq$ MaxIItn do
    Define $\widehat{\mu}_i = \begin{cases} ||c(x_i)||/||y_i - y_k^E||, & \text{if } ||y_i - y_k^E|| \neq 0 \\ \text{Inf}, & \text{otherwise} \end{cases}$
                                                                    ▷ Possibly increase $\mu_i$
    Define $\mu_i = \begin{cases} \max(\widehat{\mu}_i, \mu_{i-1}), & \text{if } \widehat{\mu}_i \leq \frac{1}{k} \\ \mu_{i-1}, & \text{otherwise} \end{cases}$
    Choose a symmetric $H_i \succ 0$, s.t. $\text{In}(K_i) = \text{In}\begin{pmatrix} H_i & J_i^T \\ J_i & -\mu_i I \end{pmatrix} = (n, m, 0)$;
    Compute the Newton direction by solving $K_i(p_i, -q_i) = -F(x_i, y_i; y_k^E, \mu_i)$;
    The search direction is given by $d_i = (p_i, q_i)$;
    Set initial step $\alpha_i \leftarrow 1$;
    while $\mathcal{M}_{\mu_i}(v_i + \alpha_i d_i) > \mathcal{M}_{\mu_i}(v_i) + \eta_s \alpha_i \nabla \mathcal{M}_{\mu_i}(v_i)^T d_i$ do
        $\alpha_i \leftarrow \gamma_c \alpha_i$;                                    ▷ Armijo line search
    end while
    $x_{i+1} \leftarrow x_i + \alpha_i p_i$;
    $y_{i+1} \leftarrow y_i + \alpha_i q_i$;
    IItn $\leftarrow$ IItn + 1;
end while
```

Example 3.4.1. *Consider applying the primal-dual path-following augmented Lagrangian method Algorithm 3.3 and Algorithm 3.4 to solve problem HS7 again.*

The path-following trajectories from different starting points are shown in Figure 3.3. The level curves of the objective function are shown in colors with each function value indicated. It can be observed that initially, the merit function $\mathcal{M}(x, y; y_k^E, \mu)$ is minimized to force



(a) $x_0 = (1, 0)$

(b) $x_0 = (1.5, -2)$

(c) $x_0 = (-2, 8)$

(d) $x_0 = (-2, 0)$

Figure 3.3: Path-Following Trajectories of Primal-Dual Path-Following Augmented Lagrangian Method

the iterates into a neighborhood of x^*, then the iterates become the Newton iterates in the limit and no inner iterations are needed. In the above figure, the final iterates converge to x^* in the same "top down" manner. The numerical results show that local quadratic convergence rate can be obtained in this example. □

3.5 Convergence Analysis

In this section, the convergence results of both outer and inner iterations are provided. Under certain mild conditions, it can be shown that in inner iterations, either

$\liminf_{i\to\infty} ||F(x_i, y_i; y^E_k, \mu_i)|| = 0$ or the sequence $\{f(x_k)\}_{k\geq 0}$ is unbounded from below. The iterates in outer iterations will converge to points satisfying the first order optimality conditions if the sequence of Lagrange multipliers $\{y_k\}_{k\geq 0}$ are bounded and the primal iterates are assumed to be within a compact set.

Theorem 3.5.1. *In Algorithm 3.4, if the direction $(p_i, -q_i)$ computed from Step 6 satisfies $K_i(p_i, -q_i) = -F(x_i, y_i; y^E_k, \mu_i)$, where K_i satisfies the condition in Step 5, then the search direction $d_i = (p_i, q_i)$ is a descent direction for $\mathcal{M}_{\mu_i}(x, y)$ at (x_i, y_i).* □

Proof. The equivalent Newton step for finding $\nabla\mathcal{M}_{\mu_i}(x, y) = 0$ is given by

$$\begin{pmatrix} H(x_i, y_i) & J(x_i)^T \\ J(x_i) & -\mu_i I_m \end{pmatrix} \begin{pmatrix} p_i \\ -q_i \end{pmatrix} = -\begin{pmatrix} g(x_i) - J(x_i)^T y_i \\ \mu_i(y_i - \pi_i) \end{pmatrix}. \tag{3.22}$$

A rearrangement gives

$$\begin{aligned} \nabla\mathcal{M}_{\mu_i}(x_i, y_i)^T(p_i, q_i) &= g(x_i)^T p_i - (2\pi_i - y_i)^T J(x_i) p_i + \mu_i (y_i - \pi_i)^T q_i \\ &= -\frac{2}{\mu_i}(J(x_i)p_i + \mu_i q_i)^T J(x_i) p_i - \mu_i ||q_i||^2 - p_i^T H(x_i, y_i) p_i \\ &= -p_i^T \left(H(x_i, y_i) + \frac{1}{\mu_i} J(x_i)^T J(x_i) \right) p_i - \frac{1}{\mu_i} ||J_i p_i + \mu_i q_i||_2^2. \end{aligned} \tag{3.23}$$

The choice of $H(x_i, y_i)$ requires $H(x_i, y_i) + \frac{1}{\mu_i} J(x_i)^T J(x_i)$ to be positive definite, so for any $d_i \neq 0$ it holds that $\nabla\mathcal{M}_{\mu_i}(x_i, y_i)^T d_i < 0$, which implies that d_i must be a descent direction for $\mathcal{M}_{\mu_i}(x, y)$ at (x_i, y_i). □

Given that d_i is a descent direction for $\mathcal{M}_{\mu_i}(x, y)$ at (x_i, y_i), the Armijo line-search can be performed along d_i to obtain sufficient decrease on $\mathcal{M}_{\mu_i}(x_i, y_i)$ once the unit step is rejected. The convergence result of the inner iterations is summarized in the following Theorem 3.5.2.

Theorem 3.5.2. *Assume that for all i, it holds that $J_i = J(x_i)$ and $H_i = H(x_i, y_i)$ are bounded and that $H_i + \frac{1}{\mu_i} J_i^T J_i$ is uniformly positive definite, i.e., $p^T(H_i + \frac{1}{\mu_i} J_i^T J_i) p \geq \lambda_s ||p||^2$*

for some $\lambda_s > 0$ and for all $p \neq 0$. Then for any fixed y_k^E, the inner iterates converge in the sense that

$$\liminf_{i\to\infty} ||F(x_i, y_i; y_k^E, \mu_i)|| = 0 \;\; or \;\; \lim_{i\to\infty} f(x_i) = -\infty. \tag{3.24}$$

Proof. The proof is by contradiction in three parts. Assume that (3.24) fails, i.e., $\{f(x_i)\}_{i\geq 0}$ are bounded below and also $\liminf_{i\to\infty} ||F(x_i, y_i; y_k^E, \mu_i)|| > 0$. According to (3.22), since H_i, J_i are bounded and $\mu_i > 0$ is bounded above by some $\bar{\mu}$, and on the right hand side $\liminf_{i\to\infty} ||F(x_i, y_i; y_k^E, \mu_i)|| > 0$, so there must exist an $\varepsilon > 0$, such that $||d_i|| \geq \varepsilon$ for all i.

The first part is to show that there exists a positive constant θ, such that

$$-\nabla \mathcal{M}_{\mu_i}(v_i)^T d_i \geq \theta ||d_i||^2 \quad \text{for all } i. \tag{3.25}$$

Suppose that (3.25) does not hold, then there must exist a subsequence $\{i_j\}_{j\geq 0}$ such that

$$-\nabla \mathcal{M}_{\mu_{i_j}}(v_{i_j})^T d_{i_j} < \theta_{i_j} ||d_{i_j}||^2, \tag{3.26}$$

for some positive sequence $\theta_{i_j} \to 0$ as $j \to \infty$. Then according to (3.23) it must hold that

$$0 \leq p_{i_j}^T (H_{i_j} + \frac{1}{\mu_{i_j}} J_{i_j}^T J_{i_j}) p_{i_j} + \frac{1}{\mu_{i_j}} ||J_{i_j} p_{i_j} + \mu_{i_j} q_{i_j}||_2^2 < \theta_{i_j} ||d_{i_j}||^2. \tag{3.27}$$

and

$$0 \leq \lambda_s ||p_{i_j}||^2 + \frac{1}{\mu_{i_j}} ||J_{i_j} p_{i_j} + \mu_{i_j} q_{i_j}||_2^2 < \theta_{i_j} ||d_{i_j}||^2. \tag{3.28}$$

The relation (3.28) is obtained from the uniform positive definiteness of H_i. Let $j \to \infty$ in (3.28), then it can be concluded that $\lim_{j\to\infty} ||p_{i_j}|| = 0$, which further yields that $\lim_{j\to\infty} ||q_{i_j}|| = 0$. So $\lim_{j\to\infty} ||d_{i_j}|| = 0$, which is a contradiction to $||d_i|| \geq \varepsilon$ for all i. Thus (3.25) holds.

The second part is to show that $\lim_{i\to\infty} v_i = v^*$ and $\lim_{i\to\infty} \alpha_i = 0$. In the inner iteration, denote $v_{i+1} = v_i + \alpha_i d_i$. The Armijo condition (Step 9 in Algorithm 3.4) gives

$$-\eta_s \alpha_i \nabla \mathcal{M}_{\mu_i}(v_i)^T d_i \leq \mathcal{M}_{\mu_i}(v_i) - \mathcal{M}_{\mu_i}(v_{i+1}) \leq \mathcal{M}_{\mu_i}(v_i) - \mathcal{M}_{\mu_{i+1}}(v_{i+1}). \tag{3.29}$$

The latter inequality is based on $\mu_{i+1} \geq \mu_i$ and that the merit function $\mathcal{M}_\mu(v)$ has a smaller value with larger μ at the same v. Since $-\nabla \mathcal{M}_{\mu_i}(v_i)^T d_i \geq \theta||d_i||^2 \geq \theta\varepsilon||d_i||$, it can be concluded that for any $N > 0$, it holds that

$$\begin{aligned}\sum_{i=1}^{N} \eta_s \theta \varepsilon ||\alpha_i d_i|| &\leq \sum_{i=1}^{N} (\mathcal{M}_{\mu_i}(v_i) - \mathcal{M}_{\mu_{i+1}}(v_{i+1})) \\ &= \mathcal{M}_{\mu_1}(v_1) - \mathcal{M}_{\mu_{N+1}}(v_{N+1}) \\ &\leq \mathcal{M}_{\mu_1}(v_1) - f(x_{N+1}) < \infty.\end{aligned} \tag{3.30}$$

It is obvious that $f(x_{N+1}) < \mathcal{M}_{\mu_{N+1}}(v_{N+1})$ from the definition of $\mathcal{M}_\mu(v)$. By the assumption that $\{f(x_i)\}_{i\geq 0}$ is bounded below, let $N \to \infty$, $\sum_{i=1}^{\infty} ||\alpha_i d_i|| = \sum_{i=1}^{\infty} ||v_{i+1} - v_i||$ is absolutely convergent. So there must exist v^*, such that $\lim_{i\to\infty} v_i = v^*$. It follows that

$$0 = \lim_{i\to\infty} ||v_{i+1} - v_i|| = \lim_{i\to\infty} ||\alpha_i d_i||. \tag{3.31}$$

As $||d_i|| \geq \varepsilon$ for all i, then $\lim_{i\to\infty} \alpha_i = 0$.

The final part completes the proof by establishing the main contradiction. Because $\lim_{i\to\infty} \alpha_i = 0$, the Armijo condition is violated, which means that there exists a $\bar{v}_i = v_i + \bar{\alpha} d_i$ with $0 < \bar{\alpha} < 1$ such that

$$\mathcal{M}_{\mu_i}(\bar{v}_i) - \mathcal{M}_{\mu_i}(v_i) > \eta_s \bar{\alpha} \nabla \mathcal{M}_{\mu_i}(v_i)^T d_i. \tag{3.32}$$

By the mean-value theorem, there exists $\widehat{v}_i = v_i + \hat{\alpha} d_i$, $0 < \hat{\alpha}_i < \bar{\alpha}_i$, such that

$$\mathcal{M}_{\mu_i}(\bar{v}_i) - \mathcal{M}_{\mu_i}(v_i) = \bar{\alpha} \nabla \mathcal{M}_{\mu_i}(\widehat{v}_i)^T d_i. \tag{3.33}$$

Condition (3.32) and (3.33) together give that

$$\nabla \mathcal{M}_{\mu_i}(\widehat{v}_i)^T d_i - \nabla \mathcal{M}_{\mu_i}(v_i)^T d_i > (\eta_s - 1) \nabla \mathcal{M}_{\mu_i}(v_i)^T d_i > 0. \tag{3.34}$$

Combining (3.25) and (3.34) yields

$$\begin{aligned}(1-\eta_s)\theta||d_i||^2 &< |(\nabla\mathcal{M}_{\mu_i}(\widehat{v}_i)-\nabla\mathcal{M}_{\mu_i}(v_i))^T d_i| \\ &\le ||\nabla\mathcal{M}_{\mu_i}(\widehat{v}_i)-\nabla\mathcal{M}_{\mu_i}(v_i)||\cdot||d_i||.\end{aligned} \tag{3.35}$$

The last inequality follows from the Cauchy-Schwarz inequality, and a simplification of (3.35) gives the inequality

$$||\nabla\mathcal{M}_{\mu_i}(\widehat{v}_i)-\nabla\mathcal{M}_{\mu_i}(v_i)|| > (1-\eta_s)\theta||d_i|| \ge (1-\eta_s)\theta\varepsilon,$$

because $\lim_{i\to\infty}||\nabla\mathcal{M}_{\mu_i}(\widehat{v}_i)-\nabla\mathcal{M}_{\mu_i}(v_i)||=0$, and $(1-\eta_s)\theta\varepsilon$ is a positive constant, which leads to a contradiction. □

The following two theorems provide the main convergence results for the outer iterations of PDAL.

Theorem 3.5.3. *Suppose the inner iterations successfully terminate, and let $\{v_k\}_{k\ge0}$ be the sequence generated by the outer iterations. Assume that both $\{||g(x_k)||\}_{k\ge0}$ and $\{||J(x_k)||\}_{k\ge0}$ are bounded. Then there are two possible outcomes.*

- *The sequence $\{y_k\}_{k\ge0}$ is unbounded, in this case, the primal iterates approach a point at which the LICQ does not hold, in other words, $\{x_k\}_{k\ge0}$ has an accumulation point x^* at which $J(x^*)$ is rank deficient,*
- *The sequence $\{y_k\}_{k\ge0}$ is bounded, in this case, $\lim_{k\to\infty}c(x_k)=c(x^*)=0$.* □

Proof. The assumption that $\{||g(x_k)||\}_{k\ge0}$ and $\{||J(x_k)||\}_{k\ge0}$ are bounded is not uncommon, for example, it can be assumed that all the iterates are within a compact set. Consider the following two cases separately.

Case 1: If $\{y_k\}_{k\ge0}$ is unbounded, then there exists a subsequence of $\{||y_k||\}_{k\ge0}$ that goes to infinity. By passing to a subsequence if necessary, it may be assumed without loss of generality that $||y_k|| \to \infty$. If $a_k = y_k/||y_k||$, $J(x_k) = J_k$, then

$$||J_k^T a_k|| \leq \frac{||g(x_k) - J_k^T y_k|| + ||g(x_k)||}{||y_k||} \to 0.$$

As $\{||J(x_k)||\}_{k\geq 0}$ is bounded and $||a_k|| = 1$ for all k, they have a limits J^* and a^* satisfying $J^* a^* = 0$. As $||a^*|| = 1$, the matrix J^* must be rank deficient.

Case 2: If $\{y_k\}_{k\geq 0}$ is bounded. By the assumption that each inner iterations successfully terminate, it holds that $\lim_{k\to\infty} ||\pi_k - y_k|| = 0$, where $\pi_k = y_k^E - c(x_k)/\mu_k$. It follows that

$$\begin{aligned} ||c(x_k)|| &= ||\mu_k(y_k^E - \pi_k)|| \\ &= |\mu_k|||y_k^E - \pi_k|| \\ &\leq |\mu_k| \left(||y_k^E - y_k|| + ||y_k - \pi_k||\right) \to 0, \end{aligned}$$

as $k \to \infty$. In this case, it must hold that $\lim_{k\to\infty} c(x_k) = c(x^*) = 0$. □

Theorem 3.5.4. *Suppose the inner iterations successfully terminate, and $\{v_k\}_{k\geq 0}$ be the sequence generated by the outer iterations. Assume further that $\{x_k\}_{k\geq 0}$ is contained in a compact set, $\{H_k\}_{k\geq 0}$ is bounded and $\left\{H_k + \frac{1}{\mu_k} J_k^T J_k\right\}_{k\geq 0}$ is uniformly positive definite. If (NEP) is feasible, then there are two possible outcomes:*

- *Case 1: $\{||y_k||\}_{k\geq 0}$ is unbounded and $\lim_{k\to\infty} x_k = x^*$ such that $J(x^*)$ is rank deficient,*
- *Case 2: $\{||y_k||\}_{k\geq 0}$ is bounded and any limit point of $\{v_k\}_{k\geq 0}$ satisfies the first-order optimality condition of (NEP).* □

Proof. By the assumption $\{x_k\}_{k\geq 0}$ is contained in a compact set, it holds that $\{||g(x_k)||\}_{k\geq 0}$ and $\{||J(x_k)||\}_{k\geq 0}$ are bounded since both the objective function $f(x)$ and the constraints $c(x)$ are assumed to be twice continuously differentiable. If $\{||y_k||\}_{k\geq 0}$ is unbounded, then according to Theorem 3.5.3, $\{||J(x_k)||\}_{k\geq 0}$ has an accumulation point J^* that is rank deficient. If $\{||y_k||\}_{k\geq 0}$ is bounded, then the second case of Theorem 3.5.3 gives that $\lim_{k\to\infty} c(x_k) = c(x^*) = 0$. Step 14 in Algorithm 3.3 implies that $\lim_{k\to\infty} ||F(x_k, y_k; y_k^E, \mu_k)||_\infty = 0$, thus the first-order optimality condition of (NEP) holds as well. □

3.6 Acknowledgement

This chapter, in part is currently being prepared for submission for publication of the material. Su, Fangyao; Gill, Philip E. The dissertation author was the primary investigator and author of this material.

Chapter 4

A Combined Trust-Region Line-Search Method

In the methods considered in the preceding chapters, it may be necessary to modify the KKT matrix to obtain the correct inertia for the line-search method. In a trust-region method, however, no such modifications are needed and thus the local quadratic model of objective function is allowed to be nonconvex. The price that has to be paid is that trust-region subproblem is generally more expensive to solve. In this chapter, background of trust-region methods will be introduced, serving as a general background for Chapter 5.

In this chapter, the combined trust-region line-search method proposed by Gill and Gertz [16] will be described. In their method, a line-search strategy is combined with a trust-region method where the sufficient decrease condition is imposed. The line-search step will also be used to update the trust-region radius. This motivates applying the method to minimize the shifted penalty-barrier function because in conventional trust-region method, a failure in obtaining a sufficient decrease in objective function generally requires the trust-region radius to be decreased until some progress can be made. But in the case of minimizing the shifted penalty-barrier function, the next step also needs to remain in the strict interior of the feasible region, so a line-search strategy to cut it back into the feasible region may save unnecessary iterations where the trust region radius is decreased. The combined strategy is used when minimizing the shifted penalty-barrier function in Chapter 5.

4.1 Background on Trust-Region Methods

Iterative methods for unconstrained optimization problem could be classified into two categories: line-search method and trust-region method. Trust-region method is robust with strong convergence property, and unlike line-search method where inertia of the Hessian matrix might need to be modified, could solve ill-conditioned problems. The local *nonconvex* quadratic models are allowed in trust-region method, which is a huge advantage over its line-search rivals.

At each iteration of trust-region method, the following trust-region subproblem (TRS) needs to be solved

$$\underset{p}{\text{minimize}}\;\; \mathcal{Q}_k(p) = g_k^T p + \tfrac{1}{2} p^T B_k p \quad \text{subject to} \quad ||p|| \le \delta_k, \tag{4.1}$$

where $\delta_k > 0$ is called the *trust-region radius*, $g_k = \nabla f(x_k)$ and B_k is either $\nabla^2 f(x_k)$ or its approximation. There is no requirement for B_k to be positive definite since a continuous quadratic function always achieves its minimum on the compact set $\overline{\mathcal{B}}(x_k; \delta_k) = \{x : ||x - x_k|| \le \delta_k\}$. A more general restriction on trust-region radius usually takes the form $||N_k p|| \le \delta_k$ where the nonsingular matrix N_k is used to scale the problem. There is no consensus on what is the best N_k and for simplicity we choose $N_k = I$ for all k, which is acceptable for well scaled problems.

A rough outline of a basic trust-region algorithm is given in Algorithm 4.1. At the current iterate x_k, the local quadratic model $\mathcal{Q}_k$ of the objective function is minimized within trust-region radius δ_k, which reflects how "trusted" the quadratic model can be to predict a change in $f(x)$. Upon finding an approximate solution p_k of the trust-region subproblem (4.1), a trial point $x_k + p_k$ is tested. If the actual reduction in $f(x)$ is within a given factor of the reduction predicted by the quadratic model, then $f(x)$ is assumed to have *sufficient decrease* and the trial point is accepted as the next iterate.

Moreover, if the reduction in $f(x)$ is larger than predicted, then the trust-region radius is increased under the assumption that within a larger trust region, an acceptable step can be obtained. If the trial point fails to give a sufficient decrease, then it is rejected and trust-region radius is reduced under the assumption that a smaller trust-region radius is necessary to provide a region in which the quadratic model is accurate.

Algorithm 4.1 Basic Trust-Region Method

```
Specify constants $0 < \eta_1 < \eta_2 < 1, \quad 0 < \eta_1 < \frac{1}{2}, \quad 0 < \gamma_2 < 1 < \gamma_3$;
Choose $x_0$, $k \leftarrow 0$, $\delta_k \leftarrow 1$;
while *not converged* do
    Compute $p_k$, an *approximate* solution of the trust-region subproblem (4.1);
    $\rho_k = (f(x_k + p_k) - f(x_k))/\mathcal{Q}_k(p_k)$;
    if $\rho_k \geq \eta_1$ then                                    ▷ Sufficient decrease condition
        *Successful iteration:* $x_{k+1} \leftarrow x_k + p_k$;
        if $\rho_k \geq \eta_2$ then
            $\delta_{k+1} \leftarrow \max\{\delta_k, \gamma_3||p_k||\}$;
        else
            $\delta_{k+1} \leftarrow \delta_k$;
        end if
    else
        $x_{k+1} \leftarrow x_k$;
        $\delta_{k+1} \leftarrow \gamma_2||p_k||$;
    end if
    $k \leftarrow k + 1$;
end while
```

Most of the computational efforts in Algorithm 4.1 are spent on finding an *approximate* solution of the trust-region subproblem (4.1). The following two lemmas given by Moré and Sorensen [32] provide the theoretical basis for solving trust-region subproblem and measuring the quality of an approximate solution.

Lemma 4.1.1. *Let δ_k be a given positive constant, a vector p_k is a global solution of the trust-region subproblem* (4.1) *if and only if $||p_k|| \leq \delta_k$ and there exists a unique $\sigma \geq 0$ such that the following two conditions hold*

$$(B_k + \sigma I)p_k = -g_k, \qquad \sigma \cdot (\delta_k - ||p_k||) = 0, \tag{4.2}$$

with $(B_k + \sigma I)$ positive semidefinite. Moreover if $(B_k + \sigma I)$ is positive definite, then the global minimizer is unique. □

Lemma 4.1.2. *Suppose the assumptions in Lemma 4.1.1 hold and let σ and p_k satisfy* (4.2) *with $B_k + \sigma I$ positive semidefinite. Then there are three possible outcomes:*

- *p_k solves minimize$_w$ $\{\mathcal{Q}(w) : ||w|| = ||p_k||\}$,*
- *if $\sigma = 0$ and $||p_k|| < \delta_k$, then p_k solves the trust-region subproblem* (4.1)*,*
- *if $\sigma > 0$ and $||p_k|| = \delta_k$, then p_k solves the trust-region subproblem* (4.1)*.*

Furthermore, if $B_k + \sigma I$ is positive definite, then p is the unique solution to the trust-region subproblem (4.1) □

Detailed proofs of Lemma 4.1.1 and Lemma 4.1.2 may be found in Sorensen [43]. Lemma 4.1.2 is important from computational standpoint because it provides the termination rules for an iterative method for solving the trust-region subproblem (4.1). An important result of Lemma 4.1.2 is stated in the next Lemma 4.1.3.

Lemma 4.1.3. *The trust-region subproblem* (4.1) *has no solution with $||p_k|| = \delta_k$ if and only if B_k is positive definite and $||B_k^{-1}p_k|| < \delta_k$.* □

Proof. If B_k is positive definite and $||B_k^{-1}p_k|| < \delta_k$, then Lemma 4.1.2 immediately gives that $B_k^{-1}p_k$ is the global solution. On the other hand, if solution p_k satisfies $||p_k|| < \delta_k$ then according to the second equation of (4.2), it must hold that $\sigma = 0$ and thus B_k is positive semidefinite. If B_k is singular, then there exists $z \neq 0$ in the null space of B_k such that $||p_k + z|| = \delta_k$, then the second part of Lemma 4.1.2 implies that $p_k + z$ will be a solution on the boundary, which is a contradiction! □

The vector z in the proof of Lemma 4.1.3 is important in computing an approximate solution to the trust-region subproblem, which will be discussed later in describing Moré and Sorensen's strategy in Section 4.4.

4.2 A Combined Trust-Region Line-Search Method

Combining trust-region method with line-search strategy is attractive to avoid repeatedly re-solving the trust-region subproblem when sufficient decrease condition fails to hold. Because finding a solution of the linear system in the trust-region subproblem can constitute a significant portion of computing time, thus performing a line search along the trust-region step can reduce the number of systems to be solved and significantly reduce the effort.

Historically, Toint [44], Nocedal and Yuan [34] employed line-search strategy in trust-region method to find a positive step α_k such that as long as $x_{k+1} = x_k + \alpha_k p_k$ gives $f(x_{k+1}) < f(x_k)$, x_{k+1} will be accepted. Thus they do not impose a sufficient decrease requirement on any step.

Gertz and Gill [15, 17] proposed a combined trust-region line-search method where the sufficient decrease condition on objective function is satisfied by forcing the iterates to satisfy Armijo-type condition, or stronger Wolfe-type condition (see Gertz [15]). This method is based on the observation that the step produced by solving the trust-region subproblem is often a suitable trial step for line-search technique. A detailed description of this method is given in Algorithm 4.2, which will be used again in Chapter 5.

Algorithm 4.2 A Combined Trust-Region Line-Search Method

```
 1: Specify constants $0 < \eta_1 < \eta_2 < \frac{1}{2}$,  $0 < \gamma_2 < 1 < \gamma_3$,  $1 \leq \nu \leq 1/\gamma_2$;
 2: Choose $x_0$, $k \leftarrow 0$, $\delta_k \leftarrow 1$;
 3: while *not converged* do
 4:     Compute $p_k$, an *approximate* solution of the trust-region subproblem (4.1);
 5:     $\rho_k = (f(x_k + p_k) - f(x_k))/\mathcal{Q}_k^-(p_k)$;
 6:     if $\rho_k \geq \eta_1$ then
 7:         *Successful iteration:*  $x_{k+1} \leftarrow x_k + p_k$;
 8:         if $\rho_k \geq \eta_2$ then
 9:             Choose $\delta_{k+1} \in [\delta_k, \max\,\{\delta_k, \gamma_3||p_k||\}]$;
10:         else
11:             $\delta_{k+1} \leftarrow \delta_k$;
12:         end if
13:     else
14:         Find the smallest $\ell$ in $\{1, 2, \cdots\}$ such that $\alpha_k = \gamma_2^{-\ell}$ satisfies the condition that
15:         $(f(x_k + \alpha_k p_k) - f(x_k))/\mathcal{Q}_k^-(p_k) \geq \eta_1$;                    ▷ Line search
16:         $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
17:         Choose $\delta_{k+1} \in [\alpha_k||p_k||, \alpha_k\nu||p_k||]$;          ▷ Adjust trust-region radius
18:     end if
19:     $k \leftarrow k + 1$;
20: end while
```

In Step 5 and Step 15 of Algorithm 4.2 where the sufficient decrease condition on

$f(x)$ is imposed, the model $\mathcal{Q}_k^-(p_k)$ is defined by

$$\mathcal{Q}_k^-(p_k) = \begin{cases} \mathcal{Q}_k(p_k), & \text{if } \; p_k^T B_k p_k < 0 \\ \\ g_k^T p_k, & \text{otherwise.} \end{cases}$$

Note that if $p_k^T B_k p_k \geq 0$, $\mathcal{Q}_k^-(p_k) = g_k^T p_k$ thus the sufficient decrease condition is stronger than the criteria used in conventional trust-region method.

The idea behind Algorithm 4.2 is that if the trust-region subproblem fails to yield a trial point that satisfies the sufficient decrease condition, then instead of remaining at the current point and reducing the trust-region radius until good progress could be made (as in Algorithm 4.1), a search is made along the direction p_k and the step is reduced until the sufficient decrease is satisfied and the next trust region is updated in accordance with the current step. The existence of steps satisfying the Armijo condition is guaranteed by the fact that p_k is either a direction of decrease or a direction of negative curvature.

4.3 Convergence of the Trust-Region Method

In this section, the first-order convergence results of both Algorithm 4.1 and Algorithm 4.2 will be provided. Under certain mild conditions, it can be shown that Algorithm 4.1 produces $\{x_k\}_{k\geq 0}$ for which $\lim_{k\to\infty} ||g(x_k)|| = 0$. Since only the first order derivative of $f(x)$ is used such results are called the *first-order convergence*. Broadly speaking, the convergence theory of trust-region method generally depends on how accurate an approximate solution of the trust-region subproblem (4.1) is. One of the weak requirement was proposed by Powell [40], who requires that the following conditions hold for some constant $\tau > 0$,

$$\mathcal{Q}(p_k) \leq -\tau ||g_k|| \min \{\delta_k, ||g_k||/||B_k||\} \quad \text{and} \quad ||p|| \leq \delta_k. \tag{4.3}$$

For computational efficiency, it is unrealistic to obtain an exact global minimizer of the trust-region subproblem. Instead, only an approximate solution of the trust-region subproblem is needed as long as certain convergence criteria are met.

A simple but important approximate solution of the trust-region subproblem could be obtained by minimizing the objective function along the gradient descent direction within trust-region radius. This solution is generally known as *Cauchy step* p_k^c, which is solution of the following problem

$$\mathcal{Q}_k(p_k^c) = \min_{p,\alpha} \left\{ \mathcal{Q}_k(p) : p = -\alpha g_k, ||p|| \leq \delta_k \right\}. \tag{4.4}$$

Thus, the Cauchy step can be expressed as $p_k^c = \alpha_k^c g_k$, where α_k^c is defined by

$$\alpha_k^c = \begin{cases} g_k^T g_k / g_k^T B_k g_k, & \text{if } g_k^T g_k / g_k^T B_k g_k \leq \delta_k / ||g_k|| \text{ and } g_k^T B_k g_k > 0; \\ \delta_k / ||g_k||, & \text{otherwise.} \end{cases}$$

The first proofs of first-order convergence for trust-region method are due to Powell [37, 38, 39] which are summarized below.

Lemma 4.3.1. *Given any norm $||\cdot||$, suppose a constant κ satisfies $||p||_2 \geq \kappa ||p||$ for all p, then the Cauchy step p_k^c satisfies the following inequality:*

$$\mathcal{Q}(p_k^c) \leq -\tfrac{1}{2}\kappa^2 ||g_k|| \min \left\{ \delta_k, ||g_k|| / ||B_k|| \right\}. \tag{4.5}$$

Proof. See Powell [37]. □

The minimum of $\mathcal{Q}_k(p)$ is at least as small as $\mathcal{Q}_k(p_k^c)$, which implies that Lemma 4.3.1 provides an upper bound on the minimum value of $\mathcal{Q}_k(p)$.

Theorem 4.3.2. *Suppose that $f : \mathcal{D} \subset \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable on the convex region $\mathcal{D}$, and let $\{x_k\}_{k\geq 0} \subset \mathcal{D}$ be a sequence of iterates generated by Algorithm 4.1. Assume further that p_k satisfies condition (4.1), $\{||B_k||\}_{k\geq 0}$ are bounded above and $f(x)$ is bounded below on $\mathcal{D}$, then either the convergence criteria is met and the iterates terminate or* $\liminf_{k\to\infty} ||g_k|| = 0$. □

Under the assumptions of Theorem 4.3.2, if the gradient $g(x)$ is uniformly continuous in some region containing all the iterates, then Algorithm 4.1 either terminates at some iterate where the convergence criteria is met or $\lim_{k\to\infty} ||g_k|| = 0$. Furthermore, Gertz [15] proves that the line-search technique in Algorithm 4.2 will not interfere with the first-order convergence properties of a conventional trust-region method, which is formally stated in the following Theorem 4.3.3.

Theorem 4.3.3. *Suppose the assumptions in Theorem 4.3.2 are satisfied, assume further that $g(x)$ is uniformly continuous in a region containing all the iterates $\{x_k\}_{k\geq 0}$. If the solution of trust-region subproblem p_k satisfies $g_k^T p_k \leq 0$, then Algorithm 4.2 will either terminate at some iterate where the convergence criteria is satisfied or* $\lim_{k\to\infty} ||g_k|| = 0$. □

Proof. See Theorem 2.2.8 in Gertz [15]. □

However, the first-order convergence $\lim_{k\to\infty} ||g(x_k)|| = 0$ is not sufficient in distinguishing between local minimizers, maximizers and saddle points simplify because of the lack of second order conditions. Moré and Sorensen [32] provide the following theorem which establishes the second order convergence results of trust-region method. This result further demonstrates that the performance of trust-region method depend on how accurately the trust-region subproblem can be solved.

Theorem 4.3.4. *Let $f : \mathcal{D} \subset \mathbb{R}^n \mapsto \mathbb{R}$ be twice continuously differentiable on the level set $\Omega = \{x : f(x) \leq f(x_0)\}$ and $\{x_k\}_{k\geq 0}$ are provided by Algorithm 4.1 with $B_k = \nabla^2 f(x_k)$. Assume there exists $\beta_1 > 0$ and $\beta_2 > 0$ such that $\mathcal{Q}_k(p_k) \leq -\beta_1 |\mathcal{Q}_k^*|$ with $||p_k|| \leq \beta_2 \delta_k$, where $\mathcal{Q}_k^*$ is optimal value of the trust-region subproblem (4.1). If Ω is compact then either the algorithm terminates at x_ℓ with $\nabla f(x_\ell) = 0$ and $\nabla^2 f(x_\ell)$ positive semidefinite, or $\{x_k\}_{k\geq 0}$ has a limit point $x^* \in \Omega$ with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ positive semidefinite.* □

Proof. If $\nabla f(x_\ell) = 0$ and $\nabla^2 f(x_\ell)$ is positive semidefinite for some $x_\ell \in \Omega$, then the algorithm terminates. Otherwise $\mathcal{Q}_k(p_k) < 0$ for all $k \geq 0$ and $\{x_k\}_{k\geq 0} \in \Omega$ are well defined. Suppose

p_k is the solution of the trust-region subproblem (4.1), then

$$\left(\nabla^2 f(x_k) + \sigma_k I\right) p_k = -g_k, \tag{4.6}$$

with $\sigma_k \geq 0$, $\nabla^2 f(x_k) + \sigma_k I$ positive semidefinite and $\sigma_k \cdot (\delta_k - ||p_k||) = 0$. Assume the Cholesky decomposition of $\nabla^2 f(x_k) + \sigma_k I$ is given by $\nabla^2 f(x_k) + \sigma_k I = R_k^T R_k$. To prove the convergence result, the following inequality is needed

$$\mathcal{Q}_k(p_k) \leq -\tfrac{1}{2}\beta_1 \left(||R_k p_k||^2 + \sigma_k \delta_k^2\right), \tag{4.7}$$

where $\beta_1 > 0$ is constant. Inequality (4.7) can be established as follows:

$$\begin{aligned} \mathcal{Q}_k^* &= g_k^T p_k + \tfrac{1}{2} p_k^T \nabla^2 f(x_k) p_k \\ &= -p_k^T \left(\nabla^2 f(x_k) + \sigma_k I\right) p_k + \tfrac{1}{2} p_k^T \nabla^2 f(x_k) p_k \\ &= -\tfrac{1}{2} p_k^T \nabla^2 f(x_k) p_k - \sigma_k p_k^T p_k \\ &= -\tfrac{1}{2} ||R_k p_k||^2 - \sigma_k \delta_k^2. \end{aligned}$$

As $\mathcal{Q}_k(p_k) \leq -\beta_1 |\mathcal{Q}_k^*|$, then (4.7) follows. Furthermore, it is immediate that

$$f(x_k) - f(x_{k+1}) \geq \tfrac{1}{2}\eta_s \beta_1 \left(||R_k p_k||^2 + \sigma_k \delta_k^2\right). \tag{4.8}$$

Based on the inequalities above, it can be shown that the sequence $\{\sigma_k\}_{k\geq 0}$ is not bounded away from 0 by contradiction. Assume not, if $\sigma_k \geq \varepsilon > 0$ for some constant $\varepsilon > 0$, then

$$\mathcal{Q}_k(p_k) \leq -\tfrac{1}{2}\beta_1 \sigma_k \delta_k^2 \leq -\tfrac{1}{2}\beta_1 \sigma_k ||p_k||^2/\beta_2^2 \leq -\tfrac{1}{2}\beta_1 \varepsilon ||p_k||^2/\beta_2^2. \tag{4.9}$$

The Taylor expansion of $f(x)$ gives that

$$||f(x_k + p_k) - f(x_k) - \mathcal{Q}_k(p_k)|| \leq \tfrac{1}{2}||p_k||^2 \max_{0\leq\xi\leq 1} ||\nabla^2 f(x_k + \xi p_k) - \nabla^2 f(x_k)||, \tag{4.10}$$

then (4.9) and (4.10) together show that

$$|\rho_k - 1| \leq \frac{\beta_2^2}{\beta_1 \varepsilon} \max_{0\leq\xi\leq 1} ||\nabla^2 f(x_k + \xi p_k) - \nabla^2 f(x_k)||,$$

where $\rho_k = (f(x_k + p_k) - f(x_k))/\mathcal{Q}_k(p_k)$. Inequality (4.8) implies that $\{\delta_k\}_{k\geq 0}$ converges to zero and thus $\{p_k\}_{k\geq 0}$ converges to zero as well, which implies that $\rho_k > \eta_e$ for sufficient large k and then the updating rule for the trust-region radius δ_k implies that δ_k is bounded away from zero, which is a contradiction!

So far it has been established that $\{\sigma_k\}_{k\geq 0}$ is not bounded away from zero. Thus there must exist a subsequence of $\{\sigma_k\}_{k\geq 0}$ converging to 0. As Ω is compact, it can be assumed without loss of generality that the same sequence of $\{x_k\}_{k\geq 0}$ converges to x^* in Ω. Since $\nabla^2 f(x_k) + \sigma_k I$ is positive semidefinite, $\nabla^2 f(x^*)$ must also be positive semidefinite. It can also be shown that

$$||R_k p_k||^2 \geq \frac{||\nabla f(x_k)||^2}{||\nabla^2 f(x_k)|| + \sigma_k}. \tag{4.11}$$

The inequality (4.8) implies that $lim_{k\to\infty}||R_k p_k|| = 0$, so by inequality (4.11), it can be concluded that $\nabla f(x^*) = 0$. □

Gertz [15] shows that the combined trust-region line-search method described in Algorithm 4.2 does not interfere with the second order convergence property. This result will be summarized in the following theorem, whose proof can be referred in Theorem 3.2.2 and Theorem 3.1.4 in [15], which is similar to the proof of Theorem 4.3.4.

Theorem 4.3.5. *Let $\{x_k\} \subset \Omega$ be the sequence of iterates generated by Algorithm 4.2. Assume that all iterates lie in a compact set. Let $f(x)$ be twice continuously differentiable and $\nabla f(x), \nabla^2 f(x)$ be uniformly continuous in Ω. Assume that $\lim_{k\to\infty} ||B_k - \nabla^2 f(x)|| = 0$ and that $||B_k||$ are bounded above. Assume further that there exists $\beta_2 > 0$ such that $\mathcal{Q}_k(p_k) \leq -\beta_1|\mathcal{Q}_k^*|$ and the step length α_k satisfies the condition $(f(x_k + \alpha_k p_k) - f(x_k))/\mathcal{Q}_k^-(p_k) \geq \eta_1$ in Algorithm 4.2 with $\eta_1 < \frac{1}{2}(1 - \sqrt{1-\beta_1})$. If x^* is an isolated limit point of $\{x_k\}_{k\geq 0}$, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.* □

The next section will focus on methods for finding an approximate solution of the trust-region subproblem, the subscript k will be omitted for simplicity.

4.4 Computing a Trust-Region Step

In this section, the strategy proposed by Moré and Sorensen [32] for solving the trust-region subproblem will be described, and the combined trust-region line-search method based on this strategy will be used to minimize the primal-dual shifted penalty-barrier function in Chapter 5.

Moré and Sorensen proposed an algorithm for the solution of trust-region subproblem (4.1) which is guaranteed to obtain a nearly optimal solution in a finite number of iterations. Their method aims at finding an approximate solution p satisfying

$$\mathcal{Q}(p) - \mathcal{Q}^* \leq \sigma_1(2-\sigma_1)\max\left\{|\mathcal{Q}^*|, \sigma_2\right\}, \quad ||p|| \leq (1+\sigma_1)\delta,$$

where σ_1 and σ_2 are constants in $(0,1)$, δ is the trust-region radius.

According to Lemma 4.1.3, the solution of trust-region subproblem (4.1) is straightforward if it does not lie on the boundary of the trust-region. Now assume that (4.1) has a solution on the boundary, let λ_n be the smallest eigenvalue of the Hessian matrix B, and $S_{\lambda_n} = \{z : Bz = \lambda_n z\}$ be the corresponding eigenspace. If g is not perpendicular to S_{λ_n}, then $||p_\sigma|| - \delta = 0$ has a solution $\sigma \in (-\lambda_n, \infty)$, where $p_\sigma = -(B+\sigma I)^{-1}g$.

This solution can be found by solving the secular equation defined in (4.13) or its modified version using Newton's method. If g is perpendicular to S_{λ_n}, then $||p_\sigma||-\delta = 0$ may still have a solution on the interval $(-\lambda_n, \infty)$ and in this case the solution can be obtained by solving the secular equation. However, if $||p_\sigma|| - \delta = 0$ fails to have a solution on $(-\lambda_n, \infty)$, this leads to certain numerical difficulties, which is called the *hard case.* In the hard case, Moré and Sorensen propose to first obtain a direction p within $||p|| < \delta$ by solving

$$(B - \lambda_n I)p = -g,$$

and then determine an eigenvector $z \in S_{\lambda_n}$ satisfying

$$(B - \lambda_n I)(p + \tau z) = -g, \quad ||p + \tau z|| = \delta.$$

Then, according to Lemma 4.1.3, $p + \tau z$ will be a solution to (4.1). In practice, for computational efficiency, the eigenvector z may not necessarily be computed according to the following lemma.

Lemma 4.4.1. *Assume $\eta \in (0, 1)$ is a given constant and further assume that*

$$B + \sigma I = R^T R, \quad (B + \sigma I)p = -g, \quad \sigma \geq 0.$$

If the vector z satisfies

$$||p + z|| = \delta, \quad ||Rz||^2 \leq \eta(||Rp||^2 + \sigma\delta^2), \tag{4.12}$$

then it holds that

$$\mathcal{Q}(p + z) \leq -\tfrac{1}{2}(1 - \eta)(||Rp||^2 + \sigma\delta^2) \leq -(1 - \eta)|\mathcal{Q}^*|,$$

where $|\mathcal{Q}^*|$ is the optimal value of the trust-region subproblem (4.1) □

Proof. See Moré and Sorensen [32]. □

Lemma 4.4.1 is important in dealing with the hard case, i.e., when $\sigma \geq 0$ with $B + \sigma I$ positive definite and the solution p of $(B + \sigma I)p = -g$ satisfies $||p|| < \delta$. It states that as long as z satisfies $||Rz||^2 \leq \eta(||Rp||^2 + \sigma\delta^2)$, then $|\mathcal{Q}(p + z) - \mathcal{Q}^*| \leq \eta|\mathcal{Q}^*|$ and $p + z$ is a nearly optimal solution to the trust-region subproblem (4.1). More precisely, we attempt to find a vector $z = \tau\widehat{z}$ that satisfies condition (4.12) and $||\widehat{z}|| = 1$. There are two choices of τ, and once $\widehat{z}$ is found, the one with smaller magnitude is chosen because we aim to minimize $||Rz||$ according to (4.12). Details of how to find such $\widehat{z}$ could be referred to Sorensen [32], which uses the LINPACK technique for estimating the smallest singular value of a triangular matrix R (see [6]). Lemma 4.4.1 is thus important in setting one of the terminating criteria for trust-region subproblem, i.e., for some constant γ_1, whenever $||p|| < \delta$ and

$$||Rz|| \leq \gamma_1(1 - \gamma_1) \max\left\{\gamma_2, ||Rp||^2 + \sigma\delta^2\right\},$$

then the algorithm terminates with $p + z$ as the approximate solution.

In the non-hard case, the so-called *secular equation* was initially introduced to find σ in (4.1), which is defined by

$$\psi(\sigma) = ||p_\sigma|| - \delta. \tag{4.13}$$

The zero $\hat{\sigma}$ of $\psi(\sigma)$ on the interval $(-\lambda_n, \infty)$ needs to be found and $\sigma = \max\{0, \hat{\sigma}\}$ is the unique value of σ. the properties of secular equation are summarized in the following Lemma 4.4.2.

Lemma 4.4.2. *Define $\psi(\sigma) = ||p_\sigma|| - \delta$, where p_σ is the solution of $(B + \sigma I)p_\sigma = -g$. Let λ_n be the smallest eigenvalue of B, then $\psi(\sigma)$ is nonincreasing and convex on $(-\lambda_n, \infty)$. Furthermore, if $g \neq 0$, then $\psi(\sigma)$ is strictly decreasing and strictly convex on $(-\lambda_n, \infty)$.* □

However $\psi(\sigma)$ has singularities at the subset of eigenvalues of $-B$ which makes $\psi(\sigma)$ highly nonlinear at points close to $\hat{\sigma}$ which prevents Newton's method from finding the zeros of $\psi(\sigma) = 0$ efficiently. To avoid these numerical difficulties, a more adequate strategy is to define $\varphi(\sigma)$ as follows, which has no singularities near $\hat{\sigma}$.

$$\varphi(\sigma) = \frac{1}{\delta} - \frac{1}{||p_\sigma||}, \quad \text{where} \quad (B + \sigma I)p_\sigma = -g. \tag{4.14}$$

One benefit of using $\varphi(\sigma)$ over $\psi(\sigma)$ is that $\varphi(\sigma)$ is usually approximately linear in a large neighborhood of $\hat{\sigma}$, thus Newton's method could be applied quite efficiently. The properties of $\varphi(\sigma)$ are summarized in the following lemma, whose proof will be given since it defines the Newton iterate for finding zeros of $\varphi(\sigma)$.

Lemma 4.4.3. *Assume $\varphi(\sigma)$ is defined in (4.14) with $g \neq 0$ and $g \notin S_{\lambda_n}$, which is the eigenspace of λ_n, then the following conclusions hold.*

- *$\varphi(\sigma)$ is twice continuously differentiable on $(-\lambda_n, \infty)$,*
- *$\varphi(\sigma)$ is strictly decreasing and strictly convex on $(-\lambda_n, \infty)$,*
- *If $lim_{\sigma \to (-\lambda_n)^+} \varphi(\sigma) > 0$, then $\varphi(\sigma)$ has a unique zero on $(-\lambda_n, \infty)$.* □

Proof. Part 1 is trivial, now suppose the eigenvalues of B are given by $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, then the second equation in (4.14) gives

$$||p_\sigma|| = ||(B + \sigma I)^{-1} g||^2 = \sum_{i=1}^{n} \left(\frac{v_i^T g}{\sigma + \lambda_i} \right)^2 .$$

If $v_n^T g \neq 0$, then

$$\lim_{\sigma \to -\lambda_n} \varphi(\sigma) = \frac{1}{\delta} > 0, \quad \lim_{\sigma \to \infty} \varphi(\sigma) = -\infty. \tag{4.15}$$

By the intermediate value theorem, there must exist at least one zero of $\varphi(\sigma)$ on $(-\lambda_n, \infty)$. For all $\sigma \in (-\lambda_n, \infty)$, the derivative and the second-order derivative of $\varphi(\sigma)$ are given by

$$\begin{aligned} \varphi'(\sigma) &= \frac{w^T d}{||d||^3} = -\frac{d^T (B + \sigma I)^{-1} d}{||d||^3} \\ \varphi''(\sigma) &= \frac{3}{||d||^5} \left(||w||^2 ||d||^2 - (w^T d)^2 \right), \end{aligned} \tag{4.16}$$

where $w = -(B + \sigma I)^{-1} d$. Cauchy-Schwartz immediate implies that $\varphi''(\sigma) \geq 0$ and equality holds if and only if w is a multiple of d. If so, w is also a multiple of g according to the second equation in (4.14). So $\varphi'(\sigma) < 0$ and $\varphi''(\sigma) > 0$ on $(-\lambda_n, \infty)$, which implies $\varphi(\sigma)$ is strictly decreasing and strictly convex on $(-\lambda_n, \infty)$. Then according to (4.15), $\varphi(\sigma)$ has a unique zero on $(-\lambda_n, \infty)$. □

Lemma 4.4.4. *Assume that $\varphi(\sigma)$ has a zero $\hat{\sigma}$ on $(-\lambda_n, \infty)$, i.e., the non-hard case, then starting from $\sigma_0 \in (-\lambda_n, \hat{\sigma})$, the Newton's iterate for calculating $\hat{\sigma}$ can be written as*

$$\sigma_{k+1} = \sigma_k + \frac{||d_k||^2}{||u_k||^2} \left(\frac{||d_k|| - \delta}{\delta} \right),$$

where $R_k^T R_k d_k = -g$, $R_k^T u_k = d_k$ and $B_k + \sigma_k I = R_k^T R_k$. □

Proof. In the proof of Lemma 4.4.3, equation (4.16) gives that

$$\begin{aligned}\sigma_{k+1} &= \sigma_k - \frac{\varphi(\sigma_k)}{\varphi'(\sigma_k)} \\ &= \sigma_k + \frac{||d_k||^2}{d_k^T(B+\sigma_k I)^{-1}d_k}\left(\frac{||d_k|| - \delta}{\delta}\right) \\ &= \sigma_k + \frac{||d_k||^2}{||R_k^{-T}d_k||^2}\left(\frac{||d_k|| - \delta}{\delta}\right) \\ &= \sigma_k + \frac{||d_k||^2}{||u_k||^2}\left(\frac{||d_k|| - \delta}{\delta}\right).\end{aligned}$$

Then the conclusion immediately follows. □

In practice, the choice of the starting point $\sigma_0 \in (-\lambda_n, \hat{\sigma})$ has to be taken care if nothing is known about λ_n and $\hat{\sigma}$. A safeguarding scheme is needed where the parameters $\lambda_L, \lambda_U, \lambda_S$ are defined such that λ_S is a lower bound on $-\lambda_n$, $[\lambda_L, \lambda_U]$ in an interval of uncertainty which contains the desired λ, and is expected to converge to $(-\lambda_n, \hat{\sigma})$ with a steadily decreasing length of the interval.

Chapter 5

A Primal-Dual Path-Following Shifted Penalty-Barrier Method

5.1 Introduction

In this section, a proposed primal-dual path-following shifted penalty-barrier method (PDPB) is described for solving the nonlinear inequality constrained optimization problem (NIP). The method has a similar structure of the primal-dual path-following augmented Lagrangian method described in Chapter 3 but has been extended to handle the nonlinear inequality constraints.

In this proposed method, the optimality conditions of (NIP) have been perturbed by the penalty and barrier parameters and Lagrange multipliers estimate, which can be seen as implicitly defining a smooth path-following trajectory towards a local minimizer of (NIP). This method has a similar“two-level” structure where in outer iteration, the Lagrange multipliers estimate are updated by the current dual variables and the penalty and barrier parameters might also be decreased to enforce the feasibility and complementarity requirements. To guarantee global convergence, a primal-dual shifted penalty-barrier function that gauges distance of current iterate to the trajectory is defined as merit function. This merit function has the property that its local minimizers lie on the path-following trajectory. Once the iterate in an outer iteration departs from this trajectory, then inner iterations are called and the merit function is minimized to force the next iterate back into the neighborhood of the trajectory.

Section 5.2 provides a general background of the shifted barrier methods and path-following methods for (NIP). The proposed algorithm PDPB is described in Section 5.3. The convergence analysis of PDPB are provided in Section 5.4. Numerical results are given in Chapter 6.

5.2 Background

5.2.1 Conventional Barrier Method

This chapter focuses on solving problem (NIP), where both $f(x)$ and $c(x)$ are assumed to be twice continuously differentiable.

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

One class of methods for solving (NIP) are *barrier methods*. For inequality constrained problems, barrier methods are motivated by minimizing a sequence of *unconstrained* functions that combine the objective function and a positively weighted "barrier" term that prevents iterates from leaving the interior of the feasible region. In these methods, given an initial iterate x_0 in the set $\mathcal{F} = \{x : c(x) > 0\}$, all subsequent iterates will also lie in $\mathcal{F}$. For this reason, these methods are also known as *interior methods*. In the 1960s, interior methods were popular for solving nonlinearly constrained problems, and the main rigorous mathematical theory associated with barrier function was established by Fiacco and McCormick, who described the relationship between the minimizers of penalty and barrier function and solutions of the original constrained problem (see Fiacco and McCormick [8]).

The simplest logarithmic barrier function associated with (NIP) can be written as

$$B(x;\mu) = f(x) - \mu \sum_{i=1}^{m} \ln c_i(x), \tag{5.1}$$

where $\mu > 0$ is called *barrier parameter.*

The logarithmic terms are well defined only at points x for which $c(x) > 0$ but becomes unbounded as x approaches points where the constraint is 0, and undefined for which $c(x) < 0$, which creates a "barrier" near boundary. The idea of conventional barrier method is to define a sequence of decreasing $\{\mu_k\}_{k\geq 0}$ with $\mu_k \to 0^+$ and for a given μ_k, $B(x;\mu_k)$ is minimized while maintaining $c(x) > 0$. The *unconstrained* local minimizer $x(\mu_k)$ of $B(x;\mu_k)$ defines a continuously differentiable path called the *barrier trajectory* or *central path.* The logarithmic barrier method can be seen as a path-following method that attempts to follow this trajectory towards the local solution of (NIP).

The properties of logarithm barrier function may be intuitively illustrated by the following example HS22 from Hock-Schittkowski test collection, which is one subclass of Constrained and Unconstrained Testing Environment (CUTEst). This example will be used repeatedly throughout the whole chapter.

Example 5.2.1. *Consider the following nonlinear inequality constrained problem HS22 in two variables, where the interior of feasible region is nonempty.*

$$\text{(HS22)} \qquad \begin{aligned} \underset{x\in\mathbb{R}^2}{\text{minimize}} \quad & (x_1-2)^2 + (x_2-1)^2 \\ \text{subject to} \quad & -x_1 - x_2 + 2 \geq 0 \\ & -x_1^2 + x_2 \geq 0. \end{aligned}$$

In this problem, the unique local (and also global) minimizer is $x^* = (1,1)$. Both constraints are active at x^*, linear independent constraint qualification (LICQ) holds, and the unique optimal Lagrange multipliers is given by $y^* = (\frac{2}{3}, \frac{2}{3})$, computed from $\nabla f(x^*) = J(x^*)^T y^*$.

The level curves of objective function and feasible region of HS22 are shown in Figure 5.1. The level curves of objective function have been shown in colors with smaller value *towards its infeasible local minimizer $\widehat{x} = (2, 1)$. The shadow gray area denotes the feasible region, and the red dot is the local (and global) constrained minimizer.*

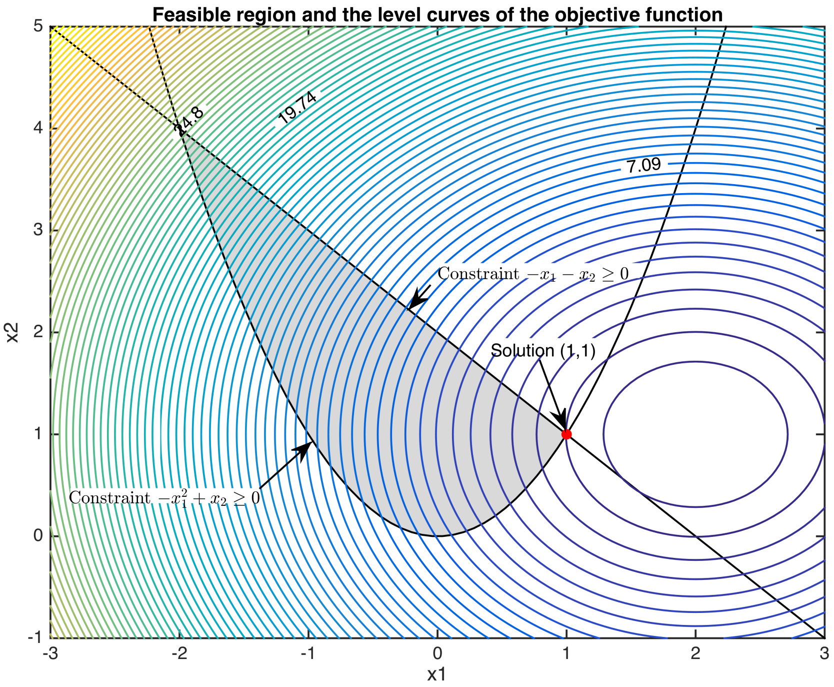


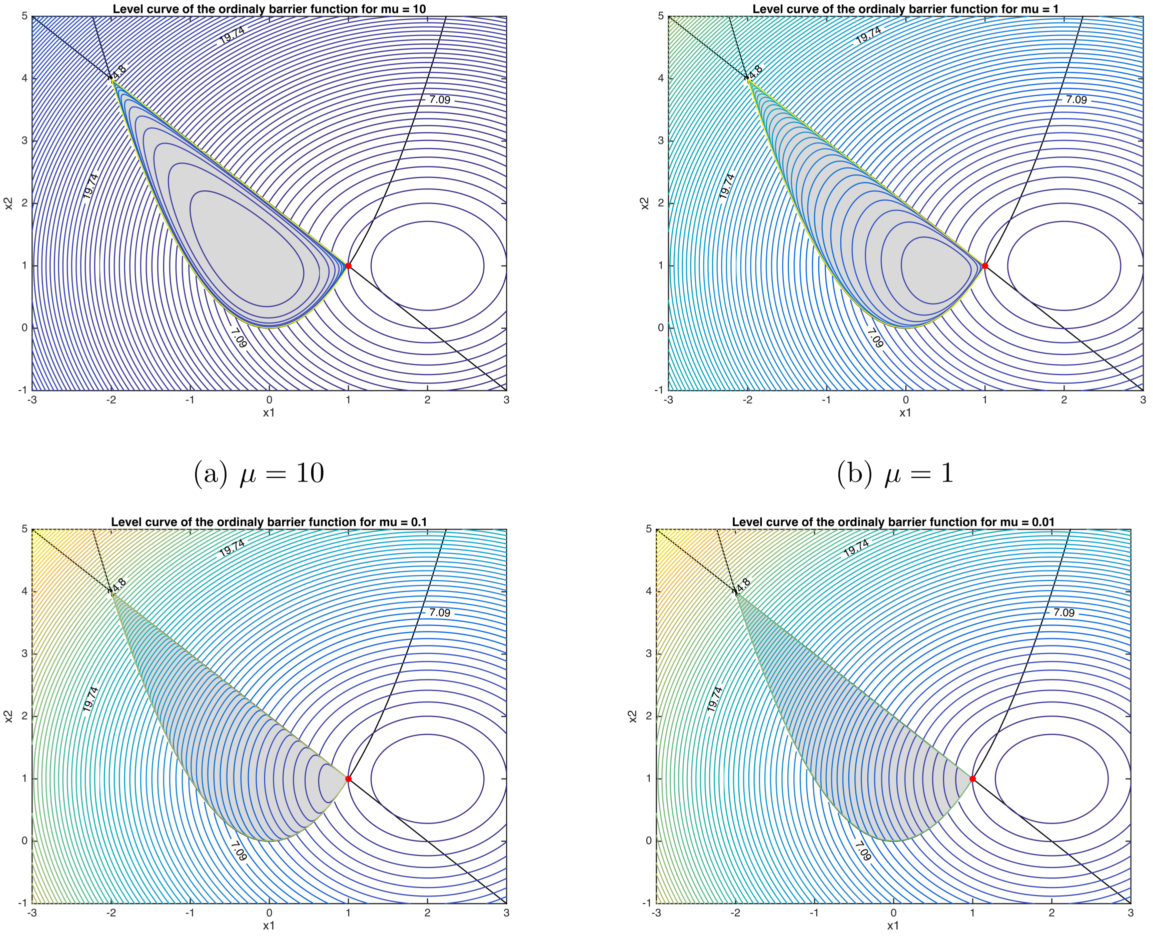
Figure 5.1: Feasible Region and Level Curves of Objective Function in HS22

The conventional logarithmic barrier function $B(x; \mu)$ in this example is given by

$$B(x; \mu) = (x_1 - 2)^2 + (x_2 - 1)^2 - \mu \ln(-x_1 - x_2 + 2) - \mu \ln(-x_1^2 + x_2).$$

The function $B(x; \mu)$ is well defined everywhere in the interior of the feasible region with the logarithmic term creates a "barrier" towards the boundary. Figures 5.2 shows the level curves of $B(x; \mu)$ for a decreasing sequence of μ.

It can be observed that as $\mu \to 0^+$, the level curves of $B(x; \mu)$ will gradually resemble the level curves of the objective function and the barrier term becomes less "weighted". The unconstrained local minimizer $x(\mu)$ of $B(x; \mu)$ will gradually approach $x^* = (1, 1)$ in the interior of the feasible region. However, on the boundary, where the barrier term tends to infinity, the level curves of $B(x; \mu)$ "jam" at x^* and μ must be reduced to zero in order for $x(\mu)$ to converge to x^*.



(a) $\mu = 10$

(b) $\mu = 1$

(c) $\mu = 0.1$

(d) $\mu = 0.01$

Figure 5.2: Level Curves of $B(x; \mu)$ in the Feasible Region for Different Values of μ.

To avoid this kind of "jam", a modified barrier method is discussed in which the barrier parameter μ is bounded away from zero. In this case, the constraints are shifted so the feasible region is enlarged to include the boundary and x^* lies in the interior of the "expanded" feasible region, see Example 5.2.2. □

If the second order sufficient conditions hold for an isolated minimizer of (NIP) (see Theorem 2.1.8), then convergence results of the barrier methods can be summarized in the following Theorem 5.2.1. To simplify the notation, the symbol $C(x)$ is used to denote the diagonalized vector $c(x)$, i.e.,

$$C(x) = \begin{pmatrix} c_1(x) & 0 & \cdots & 0 \\ 0 & c_2(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_m(x) \end{pmatrix} \tag{5.2}$$

Theorem 5.2.1 (Properties of the barrier trajectory). *Let $\{\mu_k\}$ be a strictly decreasing sequence of positive barrier parameters such that $\lim_{k\to\infty} \mu_k = 0$. Moreover, if the second order sufficient conditions (see Theorem 2.1.8) for an isolated minimizer hold at x^*, then the following conclusions hold:*

- *there is at least one subsequence of unconstrained minimizers of the barrier function $B(x; \mu_k)$ that converges to x^*;*
- *possibly passing to the sub-index, suppose $\{x_k\}_{k\geq 0}$ denote such a convergent subsequence. Then the sequence of barrier multipliers $\{y_k\}_{k\geq 0}$, defined by $y_k = \mu_k C(x(\mu_k))^{-1} e$ has the property that every of its component is bounded; and,*
- $\lim_{k\to\infty} y_k = \bar{y} \in \mathcal{Y}(x^*)$.

If in addition, strict complementarity holds at x^*, i.e., there is a vector $\bar{y} \in \mathcal{Y}(x^*)$ such that $y_i > 0$ for all $i \in \mathcal{A}(x^*)$, then the following holds:

- *$\bar{y}_a > 0$;*
- *for sufficiently large k, the Hessian matrix $\nabla^2 B(x_k, \mu_k)$ is positive definite;*
- *there exists a unique, continuously differentiable vector function $x(\mu)$ of unconstrained minimizers of $B(x, \mu)$ for positive μ in a neighborhood of $\mu = 0$; and,*
- $\lim_{\mu\to 0^+} x(\mu) = x^*$.

Proof. See Theorem 3.12 of Forsgren, Gill and Wright [13]. □

As $B(x;\mu)$ is continuously differentiable, its unconstrained local minimizer $x(\mu)$ satisfies $\nabla B(x(\mu);\mu) = 0$. The gradient and Hessian of $B(x;\mu)$ are given by

$$\nabla B(x;\mu) = g(x) - \mu J(x)^T C(x)^{-1} e$$

$$\begin{aligned} \nabla^2 B(x;\mu) &= \nabla^2 f(x) - \sum_{i=1}^{m} \frac{\mu}{c_i(x)} \nabla^2 c_i(x) + \mu J(x)^T C(x)^{-2} J(x) \\ &= H(x,\pi) + J(x)^T \mathit{\Pi}(x) C(x)^{-1} J(x), \end{aligned}$$

where $\mathit{\Pi}(x)$ is the diagonalized vector $\pi(x)$, $\pi(x) = \mu C(x)^{-1} e$, which is generally called the *barrier multipliers* (by analogy with Lagrange multipliers). Then Newton's method for finding the stationary point of $B(x;\mu)$ can be written as

$$\left(H(x,\pi) + J(x)^T \mathit{\Pi}(x) C(x)^{-1} J(x)\right) p = -\left(g(x) - J(x)^T \pi\right). \tag{5.3}$$

However, the direct unconstrained minimization of $B(x;\mu)$ is not recommended because of poor convergence as $\mu \to 0^+$. For many years the ill-conditioning of $H(x,\pi) + J(x)^T \mathit{\Pi}(x) C(x)^{-1} J(x)$ as $\mu \to 0^+$ was blamed for this poor performance. However, it now known that the real reason is that $x(\mu)$ is a poor estimate of $x(\bar{\mu})$ when μ is decreased to $\bar{\mu}$, which implies that a full Newton step cannot be taken immediately after the barrier parameter is reduced. This property implies that the classical primal barrier method is unavoidably inefficient. A more detailed analysis can be referred to Wright [45].

5.2.2 Modified Primal-Dual Interior Methods

To avoid the difficulty of minimizing $B(x;\mu)$ as $\mu \to 0^+$ in the conventional barrier method, *modified barrier method* have been introduced that defines a sequence of unconstrained problems in which the barrier parameter μ remains bounded away from zero (see Polyak [33, 35]). These methods are based on the observation that $c(x) \geq 0$ is equivalent to $\mu \ln(1 + c_i(x)/\mu) \geq 0$ for any fixed $\mu > 0$, i.e., their associated feasible sets are identical. So problem (NIP) is equivalent to the following modified problem

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & \mu \ln(1 + c_i(x)/\mu) \geq 0. \end{aligned} \tag{5.4}$$

The modified problem (5.4) serves as the main motivation for the *modified barrier function* $\mathcal{M}(x;\lambda)$ defined in the following form

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

which is the conventional Lagrangian function for problem (5.4) with λ the vector of Lagrange multipliers. An important property of $\mathcal{M}(x;\lambda)$ is that if (x^*,λ^*) is the optimal primal-dual pair of problem (NIP), then there exists a constant $\mu^* > 0$ such that for all $0 < \mu < \mu^*$, x^* is a local minimizer of $\mathcal{M}(x;\lambda^*)$, i.e., $\nabla\mathcal{M}(x^*;\lambda^*) = 0$ and $\nabla^2\mathcal{M}(x^*;\lambda^*)$ positive semidefinite (see Polyak [33]). So x^* can be obtained in just *one unconstrained* minimization if λ^* is known in advance.

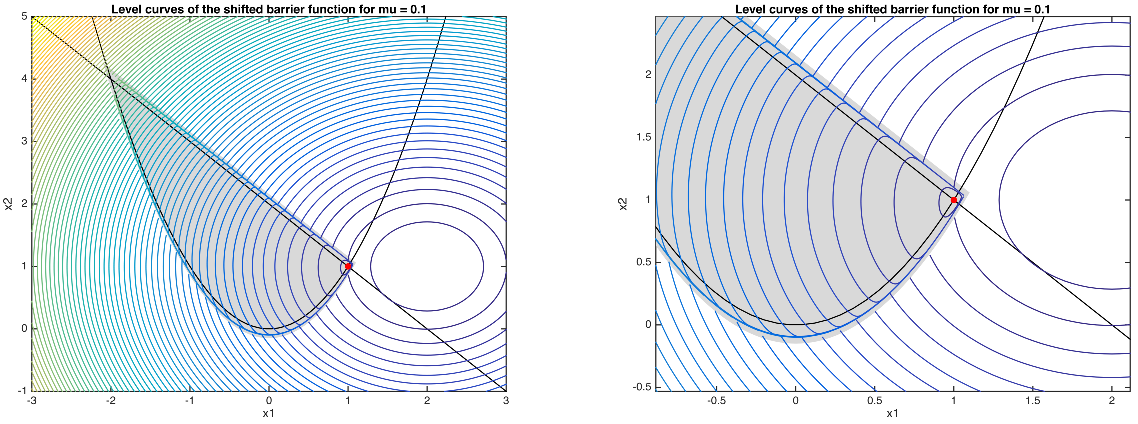
In practice, however, both u^* and λ^* are generally unknown, so a sequence of $\mathcal{M}(x;\lambda)$ defined with the estimates of u^* and λ^* needs to be minimized. The multipliers estimate is updated at each iteration and the barrier parameter is reduced once $\nabla^2\mathcal{M}(x;\lambda)$ is not sufficient positive definite, just like the updating rule for penalty parameter in Algorithm 3.3 (see also, e.g., Polyak [35]).

In defining $\mathcal{M}(x;\lambda)$, it is required that $1 + c_i(x)/\mu > 0$ for all $1 \leq i \leq m$, i.e., $c(x) + \mu e > 0$, so the feasible region has been enlarged in the sense that $c(x) = 0$ is allowed, where $c(x)$ has to be strictly positive in defining $B(x;\mu)$. To explain this intuitively, consider again the following example HS22, which has been discussed in Example 5.2.1.

Example 5.2.2. *To illustrate how the feasible region is enlarged in the modified barrier method and its benefits of keeping μ bounded away from 0, the example HS22 is reused as* *before. In this case, the modified barrier function $\mathcal{M}(x;\lambda)$ is defined by*

$$\mathcal{M}(x;\lambda) = (x_1-2)^2 + (x_2-1)^2 \\ - \mu\lambda_1 \ln(1 + (-x_1 - x_2 + 2)/\mu) - \mu\lambda_2 \ln(1 + (-x_1^2 + x_2)/\mu).$$

The feasible region and the level curves of $\mathcal{M}(x;\lambda)$ are shown below where λ is set to $\bar{\lambda} = (0.66, 0.66)$, which is the approximation of the optimal multipliers $\lambda^* = (\frac{2}{3}, \frac{2}{3})$ and the barrier parameter μ is set to 10^{-1}. The graph on the left shows the feasible region and the level curves of the modified barrier function $\mathcal{M}(x;\lambda)$ for $\mu = 10^{-1}$ and $\bar{\lambda} = (0.66, 0.66)$. The graph on the right is the "microscope" of $\mathcal{M}(x;\bar{\lambda})$ around the constrained local minimizer $x^* = (1,1)$.



(a) $\mu = 0.1, \quad \bar{\lambda} = (0.66, 0.66)$

(b) $\mu = 0.1, \quad \bar{\lambda} = (0.66, 0.66)$

Figure 5.3: Feasible Region and Level Curves of $\mathcal{M}(x;\lambda)$ with $\mu = 10^{-1}$

It can be shown that the feasible region has been expanded in the larger gray region and the level curves of $\mathcal{M}(x;\bar{\lambda})$ no longer "jam" near the boundary. In fact, the constrained local minimizer $x^* = (1,1)$ (the red point) now has been contained in the "interior" of the level curves and the barrier parameter μ need not converge to zero. □

However, equation (5.3) explicitly involves only x, the so-called *primal variables*. By far the most popular interior-point methods today are primal-dual methods, which are the next main topic.

Consider solving $\nabla B(x(\mu);\mu) = 0$ with the dual variables introduced. Denote $y(\mu) = \mu C(x(\mu))^{-1}e$, then $\nabla B(x(\mu);\mu) = 0$ can be rewritten in the following form

$$\begin{pmatrix} g(x(\mu)) - J(x(\mu))^T y(\mu) \\ c(x(\mu)) \cdot y(\mu) \end{pmatrix} = \begin{pmatrix} 0 \\ \mu e \end{pmatrix}. \tag{5.5}$$

In this form, $(x(\mu), y(\mu))$ can be regarded as tracing out the *primal-dual* trajectory as $\mu \to 0^+$, where $c(x(\mu)) > 0$ and $y(\mu) > 0$ are enforced implicitly. If we define

$$F_\mu(x,y) = \begin{pmatrix} g(x) - J(x)^T y \\ c(x) \cdot y - \mu e \end{pmatrix}, \tag{5.6}$$

then (5.5) can be regarded as a primal-dual path-following formula defined by $F_\mu(x,y) = 0$. The corresponding Newton equation for finding the zeros of $F_\mu(x,y)$ is expressed as

$$\begin{pmatrix} H(x,y) & -J(x)^T \\ YJ(x) & C(x) \end{pmatrix} \begin{pmatrix} \varDelta x \\ \varDelta y \end{pmatrix} = - \begin{pmatrix} g(x) - J(x)^T y \\ c(x) \cdot y - \mu e \end{pmatrix}, \tag{5.7}$$

where $C(x)$ is the diagonalized $c(x)$ and $H(x,y)$ is the Hessian of Lagrangian.

Naturally, the primal-dual equation (5.7) does not solely constitute a complete algorithm for nonlinear programming. The nonsingularity of the primal-dual system (5.7) can only be expected in the neighborhood of a trajectory of barrier minimizers. When the problem is nonconvex and the current primal-dual iterate is far from the trajectory, there is no guarantee that a solution of system (5.7) exists. Moreover, finding zeros of $F_\mu(x,y)$ is not enough as the second-order conditions associated with *minimizing* the barrier method have not been exploited.

For this reason and also to obtain the global convergence from any arbitrary feasible starting point, a merit function that measures and ensures the progress towards the solution (x^*, y^*) is needed. Consider the following primal-dual barrier function,

$$\mathcal{M}_\mu(x,y) = f(x) - \mu \sum_{i=1}^{m} \ln c_i(x) - \sum_{i=1}^{m} \left\{ \ln\left(\frac{y_i c_i(x)}{\mu}\right) + \left(1 - \frac{y_i c_i(x)}{\mu}\right) \right\}, \tag{5.8}$$

where $y_i > 0$ and $c_i(x) > 0$ are imposed implicitly.

The function $\mathcal{M}_\mu(x,y)$ can be seen as the classical logarithm barrier function plus a term that measures the distance of (x,y) to the trajectory $F_\mu(x,y) = 0$. If problem (NIP) is nonconvex, the advantage of using $\mathcal{M}_\mu(x,y)$ as the merit function over other functions, say, $\hat{M}_\mu(x,y) = ||F_\mu(x,y)||$ is that it can distinguish between local minimizers and maximizers by enforcing second-order conditions. Lemma 5.2.2 shows that the isolated local minimizer of the barrier function is also a local *minimizer* of $\mathcal{M}_\mu(x,y)$.

Lemma 5.2.2. *Assume that* $(x(\mu), y(\mu))$ *is an isolated local minimizer of the barrier function defined in* (5.1)*, then for any* $\mu > 0$ *sufficiently small,* $(x(\mu), y(\mu))$ *is also an unconstrained local minimizer of* $\mathcal{M}_\mu(x,y)$*.* □

Proof. To simplify the notation, denote $\pi(x) = \mu C(x)^{-1}e$, and $D(x,y) = Y^{-1}C(x)$, where Y is the diagonalized y. The gradient of $\mathcal{M}_\mu(x,y)$ can be written as

$$\nabla \mathcal{M}_\mu(x,y) = \begin{pmatrix} g(x) - J(x)^T(2\pi(x) - y) \\ D(y - \pi(x)) \end{pmatrix}.$$

If it holds that $(x,y) = (x(\mu), y(\mu))$, then $\pi(x) = \mu C(x(\mu))^{-1}e = y(\mu)$ and $g(x(\mu)) - J(x(\mu))^T y(\mu) = 0$. Since $(x(\mu), y(\mu))$ is an isolated minimizer of the barrier function $B(x;\mu)$, this implies

$$\nabla \mathcal{M}_\mu(x(\mu), y(\mu)) = \begin{pmatrix} g(x(\mu)) - J(x(\mu))^T y(\mu) \\ 0 \end{pmatrix} = 0,$$

so $(x(\mu), y(\mu))$ is a stationary point of $\mathcal{M}_\mu(x,y)$. The remainder of the proof establishes that $\nabla^2 \mathcal{M}_\mu(x(\mu), y(\mu))$ is positive definite. The Hessian of $\mathcal{M}_\mu(x,y)$ is given by

$$\nabla^2 \mathcal{M}_\mu(x,y) = \begin{pmatrix} H(x, 2\pi(x) - y) & J(x)^T \\ J(x) & D\mathit{\Pi}(x)Y^{-1} \end{pmatrix},$$

where $\mathit{\Pi}(x)$ is the diagonalized $\pi(x)$. If $(x,y)=(x(\mu),y(\mu))$, then

$$\nabla^2\mathcal{M}_\mu(x(\mu),y(\mu))=\begin{pmatrix} H(x(\mu),y(\mu)) & J(x(\mu))^T \\ J(x(\mu)) & D \end{pmatrix}.$$

As $\nabla^2\mathcal{M}_\mu(x(\mu),y(\mu))$ is symmetric, its inertia can be obtained by considering the Schur complement of D, and denote $\widehat{H}(\mu)=H(x(\mu),y(\mu))+J(x(\mu))^TD^{-1}J(x(\mu))$, then

$$\operatorname{In}\left(\nabla^2\mathcal{M}_\mu(x(\mu),y(\mu))\right)=\operatorname{In}\left(\widehat{H}(\mu)\right)+(m,0,0).$$

Since $\widehat{H}(\mu)$ is the Hessian of $B(x;\mu)$, which is positive definite on the trajectory defined by (5.5). Thus it must hold that

$$\operatorname{In}\left(\nabla^2\mathcal{M}_\mu(x(\mu),y(\mu))\right)=(n,0,0)+(m,0,0)=(n+m,0,0).$$

So $\nabla^2\mathcal{M}_\mu(x(\mu),y(\mu))$ is positive definite, which completes the proof. □

The scheme of following the trajectory defined by equation (5.6) together with using merit function $\mathcal{M}_\mu(x,y)$ in (5.8) to restrict all the iterates in the neighborhood of the trajectory has the same "two-level" structure of inner and outer iterations. The barrier parameter μ is steadily reduced in the outer iterations and $\mathcal{M}_\mu(x,y)$ is minimized for the fixed μ in the inner iterations.

However, there are still some drawbacks of this scheme, e.g., the treatment for equality constraints and the difficulty for finding a strictly feasible point, which is nontrivial. Moreover, the barrier parameter μ still needs to converge to zero thus certain numerical benefits are lost.

In the algorithm proposed in the next section, we aim at path-following a *perturbed* first-order optimality condition which allows both the penalty parameter μ^P and the barrier parameter μ^B to be bounded away from 0. A primal-dual shifted penalty-barrier merit function is used to obtain global convergence that is able to handle both equality and inequality constraints.

5.3 Description of the Proposed Algorithm

5.3.1 Algorithm Overview

The form of constraints in (NIP) can be treated either directly as $c(x) \geq 0$ or indirectly as equality and nonnegative slack variable $c(x) - s = 0$, $s \geq 0$. The use of slack variables avoids the need to find a strictly feasible initial point. The introduction of slack variables gives the equivalent problem (NIPs), which is repeated here for convenience:

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

Another benefit of using slack variables s is that strict feasibility with respect to simple bounds $s \geq 0$ is easily maintained, since a step to the boundary of a general nonlinear constraint must be calculated by iteration, leading to less efficient evaluations of constraints at infeasible points.

Under certain optimality conditions, the vector (x^*, s^*, y^*, w^*) is said to be a first-order KKT point for problem (NIPs) if the following optimality condition holds

$$c(x^*) - s^* = 0, \qquad s^* \geq 0, \tag{5.9}$$

$$g(x^*) - J(x^*)^T y^* = 0, \qquad y^* - w^* = 0, \tag{5.10}$$

$$s^* \cdot w^* = 0, \qquad w^* \geq 0, \tag{5.11}$$

where y^* and w^* are the optimal Lagrange multipliers for $c(x) - s = 0$ and $s \geq 0$ respectively, and (x, s, y, w) are called the *primal-dual pairs.*

Similar to the idea in primal-dual path-following augmented Lagrangian method for problem (NEP) in Algorithm 3.3 described in Chapter 3, consider the path-following equation (5.12). $F(x, s, y, w; y^E, w^E, \mu^P, \mu^B) = 0$ can be seen as a perturbation on the optimality conditions of (5.3.2) - (5.3.4).

$$F(x,s,y,w;y^{E},w^{E},\mu^{P},\mu^{B}) = \begin{pmatrix} g(x) - J(x)^{T}y \\ y - w \\ c(x) - s + \mu^{P}(y - y^{E}) \\ s \cdot w + \mu^{B}(w - w^{E}) \end{pmatrix} = 0, \tag{5.12}$$

where y^{E} and w^{E} are estimated Lagrange multipliers for $c(x) - s = 0$ and $s \geq 0$ respectively, $\mu^{P} > 0$ and $\mu^{B} > 0$ are the penalty and barrier parameters.

If $y - y^{E} = 0$ and $w - w^{E} = 0$, then (5.12) is exactly the same as the first order optimality condition of (NIPs). As $\mu^{P}(y - y^{E}) \to 0$ and $\mu^{B}(w - w^{E}) \to 0$, a zero (x, s, y, w) of $F(x,s,y,w;y^{E},w^{E},\mu^{P},\mu^{B})$ with $s > 0$ and $w > 0$ will gradually approximate the solution of equation (5.9) – (5.12). So the penalty parameter μ^{P} and the barrier parameter μ^{B}, together with the Lagrange multipliers estimates y^{E} and w^{E}, can be thought as implicitly defining a path-following trajectory, and the primal-dual pairs (x, s, y, w) tends to follow this trajectory towards a local constrained minimizer of (NIPs).

Unlike the conventional quadratic penalty function methods, which require $\mu^{P} \to 0$ (see Bertsekas [4]), the penalty parameter μ^{P} no longer needs to be reduced to zero as y^{E} is an increasingly accurate multipliers estimate. Similarly, the barrier parameter μ^{B} no longer needs to be reduced to zero as the conventional barrier methods do. This can lead to certain numeric benefits since in the conventional barrier methods, the barrier term will create a singularity near the boundary of the feasible region if $\mu^{B} \to 0$ (see Gill [13, 22, 23]).

In the outer iteration of PDPB, if the primal-dual pairs $v_k = (x_k, s_k, y_k, w_k)$ fall within current neighborhood of the trajectory defined by (5.12), then y^{E} and w^{E} are updated by the current dual variables y_k and w_k. μ^{P} and μ^{B} might also be adjusted according to the feasibility and complementarity violations. If, however, the current iterates departs from the path-following trajectory, then inner iteration is called where a shifted penalty-barrier merit function is minimized to restrict the iterates within current neighborhood of the trajectory. The combined trust-region line-search method introduced in Chapter 4 is used to minimize the merit function. For further details see Section 5.3.4.

5.3.2 Description of the Outer Iteration

In the outer iteration of PDPB, it stops once the residue of $||F(v_k; y_k^E, w_k^E, \mu_k^P, \mu_k^B)||$ is sufficiently close to 0. At the current iterate (x_k, s_k, y_k, w_k), if

$$||F(v_k; y_{k-1}^E, w_{k-1}^E, \mu_{k-1}^P, \mu_{k-1}^B)|| < \eta_1 ||F(v_{k-1}; y_{k-1}^E, w_{k-1}^E, \mu_{k-1}^P, \mu_{k-1}^B)||, \tag{5.13}$$

where $\eta_1 \in (0, 1)$ is a constant that controls the speed of $||F||$ converging to zero. If condition (5.13) is satisfied, then the current iterate v_k is thought to be sufficiently close to the path-following trajectory, and this iteration is designated as an O-iteration. In this case, v_k is accepted and the multiplier estimates are updated by $y_k^E = y_k$, $w_k^E = w_k$.

The necessity for updating the penalty and barrier parameter depends on the feasibility and complementarity violation. Define the first order stationary, feasibility and complementarity measure to be $\kappa_{stny}(v_k), \kappa_{feas}(v_k), \kappa_{comp}(v_k)$ respectively, i.e.,

$$\begin{aligned}
\kappa_{stny}(v_k) &= \max\left(||g_k - J_k^T y_k||, ||y_k - w_k||\right) \\
\kappa_{feas}(v_k) &= ||c_k - s_k|| \\
\kappa_{comp}(v_k) &= ||\min\ (q_1(v_k), q_2(v_k, \mu_k^B))\,||,
\end{aligned}$$

where $q_1(v_k)$ and $q_2(v_k, \mu_k^B)$ in $\kappa_{comp}(v_k)$ are defined by

$$\begin{aligned}
q_1(v_k) &= \max\ (|\min(s_k, w_k, 0)|, |s_k \cdot w_k|) \\
q_2(v_k, \mu_k^B) &= \max\ (|\min(s_k + \mu_k^B e, w_k, 0)|, |(s_k + \mu_k^B e) \cdot w_k|, \mu_k^B e)\,.
\end{aligned} \tag{5.14}$$

Given these definitions, a detailed description of the outer iteration of PDPB is given in Algorithm 5.1.

Algorithm 5.1 Primal-Dual Path-Following Shifted Penalty-Barrier Method (Outer)

```
Input: (prob);                                                     ▷ Input in the CUTEst format
Output: $(x^*, y^*, \text{converged})$;
Specify constants $0 < a_P, a_B, \eta_1, \rho_1, \rho_2, \theta_\mu, \gamma_c, \gamma_s < 1 \leq \nu$;
Specify constants $0 < \eta_s < \eta_e < \frac{1}{2}, 1 < \gamma_e$, Tol, MaxItn;
Initialize $x_0, s_0, y_0, w_0, y^E, w^E, \mu^P, \mu^B, \delta_0$;
Define $\kappa_{stny}(v_k) = \max\big(||g_k - J_k^T y_k||, ||y_k - w_k||\big)$.
Define $\kappa_{feas}(v_k) = ||c_k - s_k||$, and $\kappa_{comp}(v_k) = \max(||\min(s_k, w_k)||, ||s_k \cdot w_k||)$
while $||F_k|| >$ Tol and $k \leq$ MaxItn do
    if $||F_k|| \leq \eta_1 ||F_{k-1}||$ then                      ▷ Outer Iteration, adjust parameters
        if $\kappa_{feas}(v_k) \geq \rho_1 \kappa_{feas}(v_{k-1})$ then
            $\mu_k^P \leftarrow \min((\mu_k^P)^{(1+\theta_\mu)}, a_P \mu_k^P)$;
        end if
        if $\kappa_{comp}(v_k) \geq \rho_2 \kappa_{comp}(v_{k-1})$ then
            $\mu_k^B \leftarrow \min((\mu_k^B)^{(1+\theta_\mu)}, a_B \mu_k^B)$;
        end if
        $y_k^E \leftarrow y_k$;
        $w_k^E \leftarrow w_k$;
    else                                                ▷ Inner Iteration, minimize merit function
        Update primal-dual pairs $v_k = (x_k, s_k, y_k, w_k)$ by using either trust-region method
        or a combined trust-region line-search method to minimize the merit function.
    end if
    Update $F_k, \kappa_{stny}(v_k), \kappa_{feas}(v_k), \kappa_{comp}(v_k)$;
    $k \leftarrow k + 1$;
end while
```

During the outer iterations, in Step 8 and Step 11 of Algorithm 5.1, the residues of κ_{feas} and κ_{comp} are observed as they are used to determine if the penalty or barrier parameter needs to be reduced. Define $\rho_1, \rho_2 \in (0,1)$ to be the constants that control the rate of convergence of $\kappa_{feas}, \kappa_{comp}$ to zero. If $\kappa_{feas}(v_k) \geq \rho_1 \kappa_{feas}(v_{k-1})$, then μ_k^P is reduced to force feasibility. Meanwhile, if $\kappa_{comp}(v_k) \geq \rho_2 \kappa_{comp}(v_{k-1})$, then μ_k^B needs to be reduced to enforce on the complementarity.

In Step 9 and Step 12 of Algorithm 5.1, the strategy for decreasing μ^P either superlinearly in the form of $\mu_k^P = (\mu_k^P)^{(1+\theta_\mu)}$ or linearly in the form of $\mu_k^P = a_P \mu_k^P$, whichever is smaller, is the same as the one that was used in Algorithm 3.3 in Chapter 3.

If condition (5.13) fails to hold, then the current iterate v_k is thought to be not sufficiently close to the trajectory. In this case, inner iteration is called and a primal-dual shifted penalty-barrier merit function is minimized. Properties of this merit function are discussed further in the next section 5.3.3.

5.3.3 A Shifted Penalty-Barrier Merit Function

Following the trajectory defined in equation (5.12), a merit function is needed to measure the distance between the current iterate and this trajectory. This merit function can be formally defined by

$$\begin{aligned}\mathcal{M}(x,s,y,w;y^E,w^E,\mu^P,\mu^B) =& f(x) - (c-s)^T y^E \\ &+ \frac{1}{2\mu^P}\|c-s\|^2 + \frac{1}{2\mu^P}\|c-s+\mu^P(y-y^E)\|^2 \\ &- \sum_{i=1}^{m} \mu^B w_i^E \ln(w_i(s_i+\mu^B)^2) + \sum_{i=1}^{m} w_i(s_i+\mu^B),\end{aligned} \tag{5.15}$$

which was proposed by Gill, Kungurtsev, and Robinson (see Gill [19]), which can be seen as more generalized form of the Forsgren-Gill penalty-barrier function (see Forsgren and

Gill [11]) used for solving (NIP) defined by

$$\begin{aligned}\mathcal{M}^{\mu}(x,y) =& f(x) - \mu\sum_{i\in\mathcal{I}}\ln c_i(x) + \frac{1}{2\mu}\sum_{i\in\mathcal{E}}c_i(x)^2\\ &- \mu\sum_{i\in\mathcal{I}}\left(\ln\left(\frac{c_i(x)y_i}{\mu}\right) + \frac{\mu - c_i(x)y_i}{\mu}\right) + \frac{1}{2\mu}\sum_{i\in\mathcal{E}}\left(c_i(x)+\mu y_i\right)^2,\end{aligned}$$

where the slack variables s are not used, and the penalty and barrier parameters are chosen to be the same value.

When defining $\mathcal{M}(x,s,y,w;y^{E},w^{E},\mu^{P},\mu^{B})$ in equation (5.15), it is required that $w_i > 0$ and $s_i + \mu^{B} > 0$, for all $1 \le i \le m$. The requirement $s > 0$ in (NIPs) has been relaxed to be $S + \mu^{B} I > 0$, which can be seen as an enlarge of the feasible region like the modified barrier methods do. Lemma 5.3.1 shows that under certain second-order conditions, point (x,s,y,w) on the trajectory defined by (5.12) is a local minimizer of $\mathcal{M}(x,s,y,w;y^{E},w^{E},\mu^{P},\mu^{B})$. Thus, this path-following trajectory can be regarded as "trajectory of minimizers" of the merit function.

Lemma 5.3.1. *Assume that both $f(x)$ and $c(x)$ are twice continuously differentiable, and all the iterates satisfy $w_i > 0$ and $s_i + \mu^{B} > 0$ for all $1 \le i \le m$. Assume further that (x,s,y,w) is on the trajectory defined by (5.12) and denote $D_P = \mu^{P} I_m$ and $D_B = (S + \mu^{B} I_m)W^{-1}$, where S and W denote the diagonalized s and w respectively. If $H(x,y) + J(x)^T(D_P + D_B)^{-1}J(x)$ is positive definite, then (x,s,y,w) is an unconstrained minimizer of $\mathcal{M}(x,s,y,w;y^{E},w^{E},\mu^{P},\mu^{B})$ for any fixed pair $(y^{E},w^{E},\mu^{P},\mu^{B})$.* □

Proof. For simplicity, omit the notation $(y^{E},w^{E},\mu^{P},\mu^{B})$ below since they are all fixed. Denote the merit function $\mathcal{M}(x,s,y,w;y^{E},w^{E},\mu^{P},\mu^{B})$ simply by $\mathcal{M}$. The proof, as before, is to show that $\nabla\mathcal{M} = 0$ and $\nabla^2\mathcal{M}$ positive definite on the trajectory. First, the auxiliary vectors π^{P} and π^{B} needs to be defined as follows,

$$\begin{aligned}\pi^{P} &= y^{E} - \frac{1}{\mu^{P}}(c(x) - s)\\ \pi^{B} &= \mu^{B}(S + \mu^{B} I_m)^{-1}w^{E}.\end{aligned} \tag{5.16}$$

On the trajectory defined by (5.12), since $S + \mu^{B} I_m$ is positive diagonal matrix, then the following two relations hold,

$$\pi^{P} - y = -\frac{1}{\mu^{P}}(c(x) - s + \mu^{P}(y - y^{E})) = 0$$

$$\pi^{B} - w = -(S + \mu^{B} I_m)^{-1}(s \cdot w + \mu^{B}(w - w^{E})) = 0.$$

On the other hand, the gradient of $\mathcal{M}$ is given by

$$\nabla \mathcal{M}(x, s, y, w) = \begin{pmatrix} g(x) - J(x)^T(\pi^{P} + (\pi^{P} - y)) \\ -(y - w) + 2(\pi^{P} - \pi^{B}) \\ -\mu^{P}(\pi^{P} - y) \\ -(S + \mu^{B} I_m) W^{-1}(\pi^{B} - w) \end{pmatrix}.$$

If (x, s, y, w) is on the trajectory, then $\pi^{P} - y = 0$ and $\pi^{B} - w = 0$. Combined with the path-following equation (5.12), it holds that $\nabla \mathcal{M}(x, s, y, w) = 0$, thus (x, s, y, w) must be a stationary point for $\mathcal{M}$.

It remains to show that $\nabla^2 \mathcal{M}$ is positive definite on the trajectory. The Hessian of $\mathcal{M}$ on the trajectory is given by

$$\nabla^2 \mathcal{M}_T = \begin{pmatrix} H(x,y) + 2J(x)^T D_{P}^{-1} J(x) & -2J(x)^T D_{P}^{-1} & J(x)^T & 0 \\ -2D_{P}^{-1} J(x) & 2(D_{P}^{-1} + D_{B}^{-1}) & -I_m & I_m \\ J(x) & -I_m & D_{P} & 0 \\ 0 & I_m & 0 & D_{B} \end{pmatrix}. \tag{5.17}$$

As $\nabla^2 \mathcal{M}_T$ is symmetric, using the Schur complement of (D_{P}, D_{B}) gives that

$$\begin{aligned} \operatorname{In}\left(\nabla^2 \mathcal{M}_T\right) &= (2m, 0, 0) + \operatorname{In}\begin{pmatrix} H(x,y) + J(x)^T D_{P}^{-1} J(x) & -J(x)^T D_{P}^{-1} \\ -D_{P}^{-1} J(x) & D_{P}^{-1} + D_{B}^{-1} \end{pmatrix} \\ &= (3m, 0, 0) + \operatorname{In}\left(H(x,y) + J(x)^T (D_{P} + D_{B})^{-1} J(x)\right) \\ &= (3m + n, 0, 0). \end{aligned}$$

The last equality holds if $H(x,y)+J(x)^T(D_P+D_B)^{-1}J(x)$ is positive definite. Thus $\nabla^2\mathcal{M}_T$ must be positive definite on the path-following trajectory, and (x,s,y,w) must be an isolated local minimizer of $\mathcal{M}(x,s,y,w)$. □

Lemma 5.3.1 establishes the theoretical basis of the inner iteration of PDPB in the sense that $\mathcal{M}(x,s,y,w;y^E,w^E,\mu^P,\mu^B)$ gauges the distance between the current iterate to the path-following trajectory defined by (5.12). It follows that by minimizing $\mathcal{M}$, it is possible to restrict all iterates within current neighborhood of the trajectory.

The following Theorem 5.3.2 shows that $H(x,y)+J(x)^T(D_P+D_B)^{-1}J(x)$ is indeed positive definite in a sufficiently small neighborhood of a solution of (NIPs) that satisfies second order sufficient optimality conditions (see Theorem 2.1.11), and the strict complementarity condition holds.

Theorem 5.3.2. *The Hessian* $\nabla^2\mathcal{M}_T$ *defined in* (5.17) *is positive definite for all* $u=(x,s,y,w;y^E,w^E,\mu^P,\mu^B)$ *that are sufficiently close to* $u^*=(x^*,s^*,y^*,w^*;y^*,w^*,0,0)$*, where* (x^*,s^*,y^*,w^*) *is a solution of (NIPs) that satisfies the second order sufficient conditions and strict complementarity.* □

Proof. As in the proof of Lemma 5.3.1, it is necessary to show that $H(x,y)+J(x)^T(D_P+D_B)^{-1}J(x)$ is positive definite for all u sufficiently close to u^*. Let $H=H(x,y)$ and $J=J(x)$. As (x^*,s^*,y^*,w^*) is a solution of (NIP) for which strict complementarity and the second-order optimality conditions hold, it is immediate that

$$\begin{aligned}&\max\{s^*,w^*\}>0,\\&p^TH(x^*,y^*)p>0\ \text{ for all } p\neq 0 \text{ satisfying } J_{\mathcal{A}}(x^*)p=0,\end{aligned}\tag{5.18}$$

where $\mathcal{A}$ denotes the active set at x^* defined by $\mathcal{A}=\{i:[c(x^*)]_i=0\}=\{i:[s^*]_i=0\}$. Then it holds that

$$\begin{aligned}H+J^T(D_P+D_B)^{-1}J=H&+J_{\mathcal{A}}^T\left(\mu^PI_m+(S_{\mathcal{A}}+\mu^BI_m)W_{\mathcal{A}}^{-1}\right)^{-1}J_{\mathcal{A}}\\&+J_{\mathcal{I}}^T\left(\mu^PI_m+(S_{\mathcal{I}}+\mu^BI_m)W_{\mathcal{I}}^{-1}\right)^{-1}J_{\mathcal{I}},\end{aligned}$$

where $\mathcal{I} \cup \mathcal{A} = \{1, 2, \ldots, m\}$ and $\mathcal{I} \cap \mathcal{A} = \emptyset$. $S_{\mathcal{A}}$ and $S_{\mathcal{I}}$ are the submatrices of S containing of the row and columns from the index set $\mathcal{A}$ and $\mathcal{I}$ respectively, with a similar meaning for $W_{\mathcal{A}}$ and $W_{\mathcal{I}}$. If u and u^* are the quantities defined in this theorem, then the following relations hold

$$\lim_{u \to u^*} J(x) = J(x^*), \quad \lim_{u \to u^*} H(x, y) = H(x^*, y^*), \tag{5.19}$$

$$\lim_{u \to u^*} \left[\mu^{\scriptscriptstyle P} I_m + (S_{\mathcal{A}} + \mu^{\scriptscriptstyle B} I_m) W_{\mathcal{A}}^{-1}\right]_i^{-1} = \infty. \tag{5.20}$$

for all $1 \le i \le m$. Using the same argument as the Theorem 3.1 in [25], (5.18) and (5.19) imply that $H + J_{\mathcal{A}}^T \left(\mu^{\scriptscriptstyle P} I_m + (S_{\mathcal{A}} + \mu^{\scriptscriptstyle B} I_m) W_{\mathcal{A}}^{-1}\right)^{-1} J_{\mathcal{A}}$ is positive definite for all u sufficiently close to u^*. Combing this with the fact that $J_{\mathcal{I}}^T \left(\mu^{\scriptscriptstyle P} I_m + (S_{\mathcal{I}} + \mu^{\scriptscriptstyle B} I_m) W_{\mathcal{I}}^{-1}\right)^{-1} J_{\mathcal{I}}$ is positive semidefinite, we can conclude that $H + J^T (D_{\scriptscriptstyle P} + D_{\scriptscriptstyle B})^{-1} J$ is positive definite for all u sufficiently close to u^*, which concludes the proof. □

Lemma 5.3.1 and Theorem 5.3.2 together imply that under certain second order sufficient order conditions, by path-following the trajectory (5.12) and restrict all the iterates near the neighborhood of it using the merit function $\mathcal{M}(x, s, y, w; y^{\scriptscriptstyle E}, w^{\scriptscriptstyle E}, \mu^{\scriptscriptstyle P}, \mu^{\scriptscriptstyle B})$, we are able to trace out to a local constrained minimizer of (NIPs).

5.3.4 Description of the Inner Iteration

In this section, the combined trust-region line-search method (Algorithm 4.2) that is described in Chapter 4 is used to minimize the merit function $\mathcal{M}(x, s, y, w; \mu^{\scriptscriptstyle P}, \mu^{\scriptscriptstyle B}, y^{\scriptscriptstyle E}, w^{\scriptscriptstyle E})$, where $(\mu^{\scriptscriptstyle P}, \mu^{\scriptscriptstyle B}, y^{\scriptscriptstyle E}, w^{\scriptscriptstyle E})$ are fixed. To simplify the notation, the merit function is denoted by $\mathcal{M}(x, s, y, w)$ in this section. As before, the following auxiliary vectors are defined

$$\pi^{\scriptscriptstyle P} = \pi^{\scriptscriptstyle P}(x, s) = y^{\scriptscriptstyle E} - \frac{1}{\mu^{\scriptscriptstyle P}}(c(x) - s)$$

$$\pi^{\scriptscriptstyle B} = \pi^{\scriptscriptstyle B}(s) = \mu^{\scriptscriptstyle B}(S + \mu^{\scriptscriptstyle B} I_m)^{-1} w^{\scriptscriptstyle E}.$$

In a conventional trust-region method, the following trust-region subproblem (TRS) needs to be solved *approximately* at each iterate (x_k, s_k, y_k, w_k)

$$
\begin{aligned}
\text{(TRS)}\ \ \underset{p\in\mathbb{R}^{n+3m}}{\text{minimize}}\ \ & p^T\nabla\mathcal{M}_k + \tfrac{1}{2}p^T B_k p \\
\text{subject to}\ \ & \|p\|_{T_k} \le \delta_k,
\end{aligned}
\tag{5.21}
$$

where B_k is an approximation of $\nabla^2\mathcal{M}(x_k, s_k, y_k, w_k)$, T_k is a positive definite *diagonal* matrix to be defined later, and $\delta_k > 0$ is the trust-region radius. Define the positive definite diagonal matrices

$$
D_P = \mu_k^P I_m, \quad \text{and} \quad D_B = (S_k + \mu_k^B I_m)W_k^{-1}.
$$

Then the gradient of $\mathcal{M}(x, s, y, w)$ at (x_k, s_k, y_k, w_k) can be written as

$$
\nabla\mathcal{M}_k = \begin{pmatrix} g_k - J_k^T(\pi_k^P + (\pi_k^P - y_k)) \\ (\pi_k^P + (\pi_k^P - y_k)) - (\pi_k^B + (\pi_k^B - w_k)) \\ -D_P(\pi_k^P - y_k) \\ -D_B(\pi_k^B - w_k) \end{pmatrix},
$$

where $g_k = g(x_k)$ and $J_k = J(x_k)$. Using the same notation, the Hessian of $\mathcal{M}(x, s, y, w)$ at (x_k, s_k, y_k, w_k) is given by

$$
\nabla^2\mathcal{M}_k = \begin{pmatrix} H_k + 2J_k^T D_P^{-1} J_k & -2J_k^T D_P^{-1} & J_k^T & 0 \\ -2D_P^{-1} J_k & 2(D_P^{-1} + D_B^{-1} W_k^{-1} \Pi_k^B) & -I_m & I_m \\ J_k & -I_m & D_P & 0 \\ 0 & I_m & 0 & D_B W_k^{-1} \Pi_k^B \end{pmatrix},
\tag{5.22}
$$

where $H_k = H(x_k, \pi_k^P + (\pi_k^P - y_k))$ and $\Pi_k^B = \Pi^B(x_k)$, the diagonalized column vector π_k^B.

On the path-following trajectory (5.12), $\pi_k^B = w_k$ which implies that the Hessian approximation B_k can be chosen by setting $\pi_k^P = y_k$ and $W_k^{-1}\Pi_k^B = I$ in (5.22), which implies the approximation of $\nabla^2\mathcal{M}(x_k, s_k, y_k, w_k)$ to be

$$B_k = \begin{pmatrix} \widehat{H}_k + 2J_k^T D_{\scriptscriptstyle P}^{-1} J_k & -2J_k^T D_{\scriptscriptstyle P}^{-1} & J_k^T & 0 \\ -2D_{\scriptscriptstyle P}^{-1} J_k & 2(D_{\scriptscriptstyle P}^{-1} + D_{\scriptscriptstyle B}^{-1}) & -I_m & I_m \\ J_k & -I_m & D_{\scriptscriptstyle P} & 0 \\ 0 & I_m & 0 & D_{\scriptscriptstyle B} \end{pmatrix}, \tag{5.23}$$

where $\widehat{H}_k = H(x_k, y_k)$. In line-search method, a further modification on $\widehat{H}_k$ may be needed if B_k is not sufficiently positive definite. In trust-region method, however, no such further modification on $\widehat{H}_k$ is needed since the quadratic model of the objective function in the trust-region subproblem is allowed to be nonconvex.

The next lemma provides the theoretical basis for defining the positive definite diagonal T_k defined in (5.21) and for finding the approximate solution Δv_k of the trust-region subproblem.

Lemma 5.3.3. *Δv_k is a global solution of the trust-region subproblem* (5.21) *if and only if $\|\Delta v_k\|_{T_k} \le \delta_k$ and there exists $\sigma_k \ge 0$, such that*

$$(B_k + \sigma_k T_k)\Delta v_k = -\nabla\mathcal{M}_k, \quad \textit{and} \quad \sigma_k(\delta_k - \|\Delta v_k\|_{T_k}) = 0, \tag{5.24}$$

with $B_k + \sigma_k T_k$ positive semidefinite. For any global minimizer Δv_k, the value of σ_k is unique. Furthermore, if a global minimum is achieved for more than one Δv_k, then σ_k is independent of Δv_k. Moreover, if $B_k + \sigma_k T_k$ is positive definite, then the global solution Δv_k is also unique. □

The first equation in (5.24) together with the structure of B_k defined in (5.23) imply that the positive definite diagonal T_k may be set as

$$T_k = \begin{pmatrix} I_n & 0 & 0 & 0 \\ 0 & I_m & 0 & 0 \\ 0 & 0 & D_{\scriptscriptstyle P} & 0 \\ 0 & 0 & 0 & D_{\scriptscriptstyle B} \end{pmatrix},$$

where I_n and I_m are the identity matrices of size n and m respectively. With this choice of T_k, $(B_k + \sigma_k T_k)v_k = -\nabla \mathcal{M}_k$ can be written as

$$\begin{pmatrix} \widehat{H}_k + 2J_k^T D_P^{-1} J_k + \sigma_k I_n & -2J_k^T D_P^{-1} & J_k^T & 0 \\ -2D_P^{-1} J_k & 2(D_P^{-1} + D_B^{-1}) + \sigma_k I_m & -I_m & I_m \\ J_k & -I_m & (1+\sigma_k) D_P & 0 \\ 0 & I_m & 0 & (1+\sigma_k) D_B \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta s_k \\ \Delta y_k \\ \Delta w_k \end{pmatrix} = -\nabla \mathcal{M}_k. \tag{5.25}$$

For simplicity, let $B(\sigma_k) = B_k + \sigma_k T_k$.

However, the matrix $B(\sigma_k)$ never needs to be factorized and equation (5.25) will never be solved explicitly in order to find Δv_k. Instead, an equivalent equation could be obtained by multiplying the following L_k to both sides of equation (5.25).

$$L_k = \begin{pmatrix} I_n & 0 & -2J_k^T D_P^{-1} & 0 \\ 0 & I_m & 2D_P^{-1} & -2D_B^{-1} \\ 0 & 0 & I_m & 0 \\ 0 & 0 & 0 & I_m \end{pmatrix},$$

which gives the following equivalent *unsymmetric* system

$$\begin{pmatrix} \widehat{H}_k + \sigma_k I_n & 0 & -(1+2\sigma_k) J_k^T & 0 \\ 0 & \sigma_k I_m & (1+2\sigma_k) I_m & -(1+2\sigma_k) I_m \\ J_k & -I_m & (1+\sigma_k) D_P & 0 \\ 0 & I_m & 0 & (1+\sigma_k) D_B \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta s_k \\ \Delta y_k \\ \Delta w_k \end{pmatrix} = - \begin{pmatrix} g_k - J_k^T y_k \\ y_k - w_k \\ -D_P(\pi^P - y_k) \\ -D_B(\pi^B - w_k) \end{pmatrix}. \tag{5.26}$$

Equation (5.26) can be symmetrized by "transferring" the factor $-(1 + 2\sigma_k)$ into Δy_k and

Δw_k, which gives the following equivalent *symmetric* system:

$$\begin{pmatrix} \widehat{H}_k + \sigma_k I_n & 0 & J_k^T & 0 \\ 0 & \sigma_k I_m & -I_m & I_m \\ J_k & -I_m & -\hat{\sigma}_k D_P & 0 \\ 0 & I_m & 0 & -\hat{\sigma}_k D_B \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta s_k \\ -\Delta\widehat{y}_k \\ -\Delta\widehat{w}_k \end{pmatrix} = - \begin{pmatrix} g_k - J_k^T y_k \\ y_k - w_k \\ -D_P(\pi^P - y_k) \\ -D_B(\pi^B - w_k) \end{pmatrix}, \tag{5.27}$$

where $\hat{\sigma}_k = (1+\sigma_k)/(1+2\sigma_k)$, $\Delta\widehat{y}_k = (1+2\sigma_k)\Delta y_k$, and $\Delta\widehat{w}_k = (1+2\sigma_k)\Delta w_k$. Denote further that the KKT matrix in (5.27) to be $N(\sigma_k)$. Equation (5.27) is a symmetric and well-scaled system where $-\hat{\sigma}_k D_P$ and $-\hat{\sigma}_k D_B$ can be regarded as a regularization on $N(\sigma_k)$.

In fact, (5.27) can be solved in an equivalent equations in a smaller size by using the block elimination, which gives the following equations

$$\begin{aligned} \Delta\widehat{w}_k &= (I_m + \sigma_k\hat{\sigma}_k D_B)^{-1}\left(\sigma_k D_B(\pi^B - w_k) + \Delta\widehat{y}_k - w_k + y_k\right) \\ \Delta s_k &= D_B(\pi^B - w_k) - \hat{\sigma}_k D_B \Delta\widehat{w}_k, \end{aligned} \tag{5.28}$$

where Δx_k and $\Delta\widehat{y}_k$ satisfy the equations

$$\begin{pmatrix} \widehat{H}_k + \sigma_k I_n & -J_k^T \\ J_k & \hat{\sigma}_k(D_P + \widehat{D}_B) \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta\widehat{y}_k \end{pmatrix} = - \begin{pmatrix} g_k - J_k^T y_k \\ \eta_y \end{pmatrix}, \tag{5.29}$$

where in equation (5.29), the following auxiliary variables are defined.

$$\begin{aligned} \widehat{D}_B =& D_B(I_m + \sigma_k\hat{\sigma}_k D_B)^{-1} \\ \eta_y =& \hat{\sigma}_k\widehat{D}_B(\sigma_k D_B(\pi^B - w_k) - w_k + y_k) - D_B(\pi^B - w_k) - D_P(\pi^P - y_k), \end{aligned}$$

An equivalent symmetric system of equation (5.29) is given by

$$\begin{pmatrix} \widehat{H}_k + \sigma_k I_n & J_k^T \\ J_k & -\hat{\sigma}_k(D_P + \widehat{D}_B) \end{pmatrix} \begin{pmatrix} \Delta x_k \\ -\Delta\widehat{y}_k \end{pmatrix} = - \begin{pmatrix} g_k - J_k^T y_k \\ \eta_y \end{pmatrix}. \tag{5.30}$$

Denote the KKT matrix in (5.30) to be $K(\sigma_k)$. Then equation (5.30) together with (5.28) are the only linear systems that we indeed solve. After $\Delta\widehat{y}_k$ and $\Delta\widehat{w}_k$ is obtained, then we can compute $\mathit{\Delta} y$ and $\mathit{\Delta} w$ using the following equations

$$\mathit{\Delta}\widehat{y}_k = (1+2\sigma_k)\mathit{\Delta} y_k, \quad \text{and} \quad \mathit{\Delta}\widehat{w}_k = (1+2\sigma_k)\mathit{\Delta} w_k.$$

Instead of solving equation (5.25), which is in the $(n+3m)$ dimensional space, we can solve the equivalent symmetric KKT system (5.30) in a smaller space of dimension $(n+m)$. Thus certain numerical benefits could be obtained.

Another advantage is that when the trust-region subproblem is solved, $B(\sigma_k)$ is required to be positive semidefinite, and fortunately, the inertia of $B(\sigma_k)$ can be deduced from the inertia of $K(\sigma_k)$. The inertia relations of $\mathrm{In}(B(\sigma_k))$, $\mathrm{In}(N(\sigma_k))$ and $\mathrm{In}(K(\sigma_k))$ can be formally stated in the following Lemma 5.3.4.

Lemma 5.3.4. *Let $B(\sigma_k), N(\sigma_k)$ and $K(\sigma_k)$ be the KKT matrices defined in equation* (5.23), (5.27)*, and* (5.30)*. Denote the auxiliary $\bar{H}_k$ to be $\bar{H}_k = \widehat{H}_k + \sigma_k I_n + \frac{1}{\hat{\sigma}_k} J_k^T (D_P + \widehat{D}_B)^{-1} J_k$. Then the following inertia relations hold:*

$$\begin{aligned}
\mathrm{In}(B(\sigma_k)) &= (3m, 0, 0) + \mathrm{In}(\bar{H}_k)\\
\mathrm{In}(N(\sigma_k)) &= (m, 2m, 0) + \mathrm{In}(\bar{H}_k)\\
\mathrm{In}(K(\sigma_k)) &= (0, m, 0) + \mathrm{In}(\bar{H}_k).
\end{aligned}$$

Proof. As $B(\sigma_k)$, $N(\sigma_k)$ and $K(\sigma_k)$ are all symmetric, the proof can be done by repeatedly using the Schur complement to deduce the inertia of each of them. From equation (5.30), $K(\sigma_k)$ is defined as

$$K(\sigma_k) = \begin{pmatrix} \widehat{H}_k + \sigma_k I_n & J_k^T \\ J_k & -\hat{\sigma}_k (D_P + \widehat{D}_B) \end{pmatrix},$$

because $D_P + \widehat{D}_B$ is a positive diagonal matrix and $\hat{\sigma}_k > 0$, computing the Schur complement of the $(2,2)$ block gives

$$\begin{aligned}\operatorname{In}(K(\sigma_k)) &= (0,m,0) + \operatorname{In}\left(\widehat{H}_k + \sigma_k I_n + \frac{1}{\hat{\sigma}_k} J_k^T (D_P + \widehat{D}_B)^{-1} J_k\right) \\ &= (0,m,0) + \operatorname{In}(\bar{H}_k),\end{aligned}$$

where $\bar{H}_k = \widehat{H}_k + \sigma_k I_n + \frac{1}{\hat{\sigma}_k} J_k^T (D_P + \widehat{D}_B)^{-1} J_k$.

The same argument can be applied to obtain the inertia of $N(\sigma_k)$, which is defined as follows

$$N(\sigma_k) = \begin{pmatrix} \widehat{H}_k + \sigma_k I_n & 0 & J_k^T & 0 \\ 0 & \sigma_k I_m & -I_m & I_m \\ J_k & -I_m & -\hat{\sigma}_k D_P & 0 \\ 0 & I_m & 0 & -\hat{\sigma}_k D_B \end{pmatrix}.$$

As D_P and D_B are positive diagonal matrices and $\hat{\sigma}_k > 0$, the inertia of $N(\sigma_k)$ can be obtained by computing the Schur complement, which gives

$$\begin{aligned}\operatorname{In}(N(\sigma_k)) &= (0,2m,0) + \operatorname{In}\begin{pmatrix} \widehat{H}_k + \sigma_k I_n + \frac{1}{\hat{\sigma}_k} J_k^T D_P^{-1} J_k & -\frac{1}{\hat{\sigma}_k} J_k^T D_P^{-1} \\ -\frac{1}{\hat{\sigma}_k} D_P^{-1} J_k & \sigma_k I_m + \frac{1}{\sigma_k}(D_P^{-1} + D_B^{-1}) \end{pmatrix} \\ &= (m,2m,0) + \operatorname{In}\left(\widehat{H}_k + \sigma_k I_n + \frac{1}{\hat{\sigma}_k} J_k^T (D_P + \widehat{D}_B)^{-1} J_k\right) \\ &= (m,2m,0) + \operatorname{In}(\bar{H}_k).\end{aligned}$$

Since $\sigma_k I_m + \frac{1}{\sigma_k}(D_P^{-1} + D_B^{-1})$ is a positive definite diagonal matrix, computing its Schur complement gives the second equation.

The last part is done using the exactly same argument to obtain the inertia of $B(\sigma_k)$, which is defined by

$$B(\sigma_k) = \begin{pmatrix} \widehat{H}_k + 2J_k^T D_P^{-1} J_k + \sigma_k I_n & -2J_k^T D_P^{-1} & J_k^T & 0 \\ -2D_P^{-1} J_k & 2(D_P^{-1} + D_B^{-1}) + \sigma_k I_m & -I_m & I_m \\ J_k & -I_m & (1+\sigma_k) D_P & 0 \\ 0 & I_m & 0 & (1+\sigma_k) D_B \end{pmatrix}.$$

Since D_P and D_B are positive diagonal matrices and $\hat{\sigma}_k > 0$, the inertia of $B(\sigma_k)$ can be obtained by computing its Schur complement, which gives

$$\begin{aligned} \operatorname{In}(B(\sigma_k)) &= (2m, 0, 0) + \operatorname{In} \begin{pmatrix} \widehat{H}_k + \sigma_k I_n + \frac{1}{\hat{\sigma}_k} J_k^T D_P^{-1} J_k & -\frac{1}{\hat{\sigma}_k} J_k^T D_P^{-1} \\ -\frac{1}{\hat{\sigma}_k} D_P^{-1} J_k & \sigma_k I_m + \frac{1}{\sigma_k}(D_P^{-1} + D_B^{-1}) \end{pmatrix} \\ &= (3m, 0, 0) + \operatorname{In}\left(\widehat{H}_k + \sigma_k I_n + \frac{1}{\hat{\sigma}_k} J_k^T (D_P + \widehat{D}_B)^{-1} J_k \right) \\ &= (3m, 0, 0) + \operatorname{In}(\bar{H}_k). \end{aligned}$$

Then the conclusions follow immediately. □

Lemma 5.3.4 is useful in that there is no need to factorize $B(\sigma_k) \in \mathbb{R}^{(n+3m)\times(n+3m)}$ in order to check if it is positive semidefinite, which is required when solving the trust-region subproblem. We only need to factorize $K(\sigma_k) \in \mathbb{R}^{(n+m)\times(n+m)}$ when solving the linear equation (5.30) and the inertia of $B(\sigma_k)$ can be obtained as a byproduct.

The practical way for computing the inertia of $K(\sigma_k)$ can be done by doing $LDLT$ decomposition on $K(\sigma_k)$, i.e., $P^T K(\sigma_k) P^T = LDL^T$, and the number of negative eigenvalues of $K(\sigma_k)$ is the number of 2×2 blocks and negative 1×1 blocks of D. If $K(\sigma_k)$ has more than m negative eigenvalues, which means that $B(\sigma_k)$ is not positive semidefinite, then σ_k needs to be increased and $K(\sigma_k)$ needs to be refactorized and the linear equation (5.30) be solved again.

Algorithm 5.2 gives a complete description of the inner iteration of PDPB, corresponding to Step 16 to Step 17 of Algorithm 5.1, which applies a combined trust-region line-search method to minimize the merit function.

Algorithm 5.2 A Primal-Dual Path-following Shifted Penalty-Barrier Method (Inner)

```
while $||F_k|| > \eta_1||F_{k-1}||$ do
    Find an approximate solution $\Delta v_k$ of the trust-region subproblem (5.21);
    Set initial step $\alpha_k \leftarrow 1$;
    while $s_k + \mu_k^B e + \alpha_k \Delta s_k > 0$ or $w_k + \alpha_k \Delta w_k > 0$ fails do        ▷ Ensure feasibility
        $\alpha_k \leftarrow \gamma_s \alpha_k$;
    end while
    $\rho_k = (\mathcal{M}(v_k + \alpha_k \Delta v_k) - \mathcal{M}(v_k))/\mathcal{Q}_k^-(\alpha_k \Delta v_k)$;
    if $\rho_k \geq \eta_s$ then
        $v_{k+1} \leftarrow v_k + \alpha_k \Delta v_k$;
        $s_{k+1} \leftarrow \max\big(s_{k+1}, c_{k+1} - \mu_k^P(y_k^E + \frac{1}{2}(w_{k+1} - y_{k+1}))\big)$;        ▷ Reset slack variables
        if $\rho_k \geq \eta_e$ then
            $\delta_k \leftarrow \max(\gamma_e \delta_k, \gamma_e||\alpha_k \Delta v_k||_{T_k})$;
        end if
    else                                                    ▷ Line search along $\Delta v_k$
        $\beta_k \leftarrow 1$;
        while $(\mathcal{M}(v_k + \alpha_k \beta_k \Delta v_k) - \mathcal{M}(v_k))/\mathcal{Q}_k^-(\alpha_k \beta_k \Delta v_k) < \eta_s$ do
            $\beta_k \leftarrow \gamma_s \beta_k$;
        end while
        $v_{k+1} \leftarrow v_k + \alpha_k \beta_k \Delta v_k$;
        $s_{k+1} \leftarrow \max\big(s_{k+1}, c_{k+1} - \mu_k^P(y_k^E + \frac{1}{2}(w_{k+1} - y_{k+1}))\big)$;        ▷ Reset slack variables
        Choose $\delta_k \in [||\alpha_k \beta_k \Delta v_k||_{T_k}, \nu||\alpha_k \beta_k \Delta v_k||_{T_k}]$;        ▷ Adjust trust-region radius
    end if
end while
```

In Algorithm 5.2, Step 3 to Step 6 enforces that $s_k + \mu_k^B e > 0$ and $w_k > 0$ for all k. If either condition is violated, a line search is needed to find the step α_k that satisfies

$$\begin{aligned} s_k + \mu_k^B e + \alpha_k \Delta s_k &> 0 \\ w_k + \alpha_k \Delta w_k &> 0. \end{aligned} \tag{5.31}$$

As s_k and w_k are column vectors, the condition (5.31) is easier to implement than finding α_k such that $c(x_k + \alpha_k \Delta x_k) \geq 0$, since many unnecessary infeasible constraint evaluations may be needed. This is another advantage of introducing the slack variable s.

In Step 7, the model $\mathcal{Q}_k^-(p_k) = g_k^T p_k + \frac{1}{2}\min(0, p_k^T H_k p_k)$ is used to enforce the Armijo-type condition, where $\eta_s \in (0, \frac{1}{2})$,

$$\mathcal{M}(v_k + \alpha_k \Delta v_k) - \mathcal{M}(v_k) \leq \eta_s \mathcal{Q}_k^-(\alpha_k \Delta v_k), \tag{5.32}$$

which is the same condition that has been used in Algorithm 4.2.

If $\alpha_k \Delta v_k$ gives a sufficient decrease on $\mathcal{M}(x, s, y, w)$, then $v_{k+1} = v_k + \alpha_k \Delta v_k$ is accepted as the next iterate. Otherwise, an Armijo-type line search is used to find $\beta_k \in (0, 1)$ such that the Armijo condition

$$\mathcal{M}(v_k + \alpha_k \beta_k \Delta v_k) \leq \mathcal{M}(v_k) + \eta_s \mathcal{Q}_k^-(\alpha_k \beta_k \Delta v_k),$$

is satisfied. The existence of such β_k has been proved in Lemma 5.4.2. Thus $\alpha_k \beta_k \Delta v_k$ gives a sufficient decrease on $\mathcal{M}$. Also, the next trust-region radius is adjusted by

$$\delta_k \in \left[||\alpha_k \beta_k \Delta v_k||_{T_k}, \nu ||\alpha_k \beta_k \Delta v_k||_{T_k} \right],$$

which is based on the previous line-search step. This strategy for updating the trust-region radius is the same as the one used by Gill and Gertz (see Algorithm 4.2).

Finally, in Step 10 and Step 20 of Algorithm 5.2, the slack reset whenever the primal-dual pair v_k is updated is given by

$$s_{k+1} \leftarrow \max\left(s_{k+1}, c_{k+1} - \mu_k^P(y_k^E + \tfrac{1}{2}(w_{k+1} - y_{k+1}))\right).$$

This slack reset is especially useful to handle problems that are locally infeasible which is a challenge for nonconvex optimization problems. This result is summarized in Theorem 5.4.7.

Meanwhile, for inner iterations, we also tried applying the conventional trust-region method to minimize $\mathcal{M}(x, s, y, w)$, as is shown in Algorithm 5.3. The difference between these two methods is that in Algorithm 5.3, once $\alpha_k \mathit{\Delta} v_k$ fails to yield a sufficient decrease on $\mathcal{M}$ (Step 8, Algorithm 5.3), then trust-region radius δ_k is reduced and the trust-region subproblem is solved again until a sufficient decrease could be obtained.

Algorithm 5.3 A Primal-Dual Path-following Shifted Penalty-Barrier Method (Inner)

```
 1: while $||F_k|| > \eta_1 ||F_{k-1}||$ do
 2:     Solve the trust-region subproblem (5.21) approximately to find a direction $\mathit{\Delta} v_k$;
 3:     Set initial step $\alpha_k \leftarrow 1$;
 4:     while $s_k + \mu_k^B e + \alpha_k \mathit{\Delta} s_k > 0$ or $w_k + \alpha_k \mathit{\Delta} w_k > 0$ fails do        ▷ Ensure feasibility
 5:         $\alpha_k \leftarrow \gamma_s \alpha_k$;
 6:     end while
 7:     $\rho_k = (\mathcal{M}(v_k + \alpha_k \mathit{\Delta} v_k) - \mathcal{M}(v_k))/\mathcal{Q}_k^-(\alpha_k \mathit{\Delta} v_k)$;
 8:     if $\rho_k \geq \eta_s$ then
 9:         $v_{k+1} \leftarrow v_k + \alpha_k \mathit{\Delta} v_k$;
10:         $s_{k+1} \leftarrow \max\big(s_{k+1}, c_{k+1} - \mu_k^P(y_k^E + \frac{1}{2}(w_{k+1} - y_{k+1}))\big)$;        ▷ Reset $s_k$
11:         if $\rho_k \geq \eta_e$ then
12:             $\delta_k \leftarrow \max(\gamma_e \delta_k, \gamma_e ||\alpha_k \mathit{\Delta} v_k||_{T_k})$;
13:         end if
14:     else
15:         $\delta_k \leftarrow \gamma_c \delta_k$;        ▷ Reduce trust-region radius
16:     end if
17: end while
```

It is worthwhile to note that each time v_k is updated, the slack variable s_k needs to be reset (Step 9, Step 16 in Algorithm 5.2, and Step 10 in Algorithm 5.3) as well for efficiency. The same slack-variable reset strategy has been used by Gill et al. [24]. This slack reset implies that

$$c_{k+1} - s_{k+1} \leq \mu_k^P \left(y_k^E + \tfrac{1}{2}(w_{k+1} - y_{k+1}) \right), \tag{5.33}$$

which ensures that any limit point (x^*, s^*) of (x_k, s_k) satisfies $c(x^*) - s^* \leq 0$ if $\{y_k^E\}_{k\geq 0}$ and $\{w_{k+1} - y_{k+1}\}_{k\geq 0}$ are bounded and μ_k^P converging to 0, which is suitable to handle the locally infeasible problem. Lemma 5.3.5 below shows that the slack reset will not lead to an increase for the merit function $\mathcal{M}(x, s, y, w)$.

Lemma 5.3.5. *Define $\widehat{s}_k = c_{k+1} - \mu_k^P \left(y_k^E + \frac{1}{2}(w_{k+1} - y_{k+1}) \right)$, then the slack reset s_{k+1} in Step 20 of Algorithm 5.2, defined as $s_{k+1} = max(s_{k+1}, \widehat{s}_k)$ will not lead to an increase in the merit function $\mathcal{M}(x, s, y, w)$.* □

Proof. Since the slack reset procedure has the effect of possibly increasing the value of some of its components, which means the term $-\sum_{i=1}^m \mu^B w_i^E \ln(w_i(s_i+\mu^B)^2)$ in $\mathcal{M}$ will not increase (see the definition of $\mathcal{M}$ in (5.15)). Denote the rest part of $\mathcal{M}$ by $\widehat{\mathcal{M}}$, i.e.,

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

Then it holds that

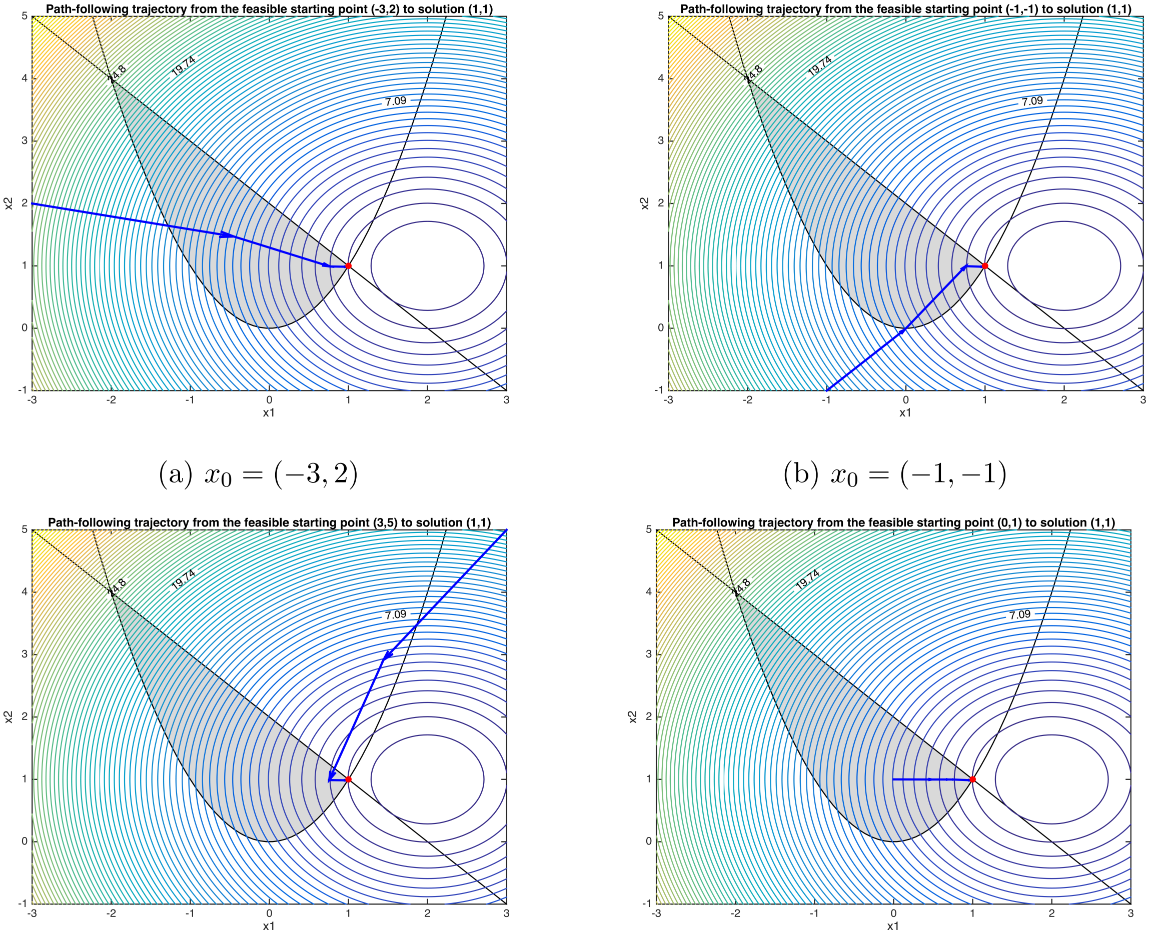
$$\begin{aligned}\nabla_s \widehat{\mathcal{M}}(x, s, y, w) &= y^E + w + \frac{1}{\mu^P}(2(s - c(x)) - \mu^P(y - y^E)) \\ \nabla_{ss} \widehat{\mathcal{M}}(x, s, y, w) &= \frac{2}{\mu^P} > 0.\end{aligned}$$

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

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

Also $\widehat{\mathcal{M}}$ is strictly convex with respect to s, so $\widehat{s}$ minimizes $\widehat{\mathcal{M}}(x,s,y,w)$. Thus after $(x_{k+1},s_{k+1},y_{k+1},w_{k+1})$ is obtained, and for fixed μ_k^P, y_k^E in inner iteration, a further slack reset $\widehat{s}_k = c_{k+1} - \mu_k^P\left(y_k^E + \frac{1}{2}(w_{k+1} - y_{k+1}\right)$ will not lead to an increase in $\mathcal{M}$. □

Example 5.3.1. *Consider problem HS22 that has been used in Example 5.2.1. The proposed primal-dual path-following shifted penalty-barrier method has been used to solve this problem again and the trajectories are shown in Figure 5.4 corresponding to different starting points.*



(a) $x_0 = (-3, 2)$

(b) $x_0 = (-1, -1)$

(c) $x_0 = (3, 5)$

(d) $x_0 = (0, 1)$

Figure 5.4: Different Path-Following Trajectories Starting From Either Feasible or Infeasible Points.

The first three graphs in Figure 5.4 show path-following trajectories associated with infeasible starting points and the last one shows the trajectory with a feasible starting point. Initially, the merit function $\mathcal{M}(x,s,y,w;y_k^E,w_k^E,\mu_k^P,\mu_k^B)$ *is minimized to drive the iterates* *into some neighborhood of x^*, then the iterates are Newton iterates in the limit and inner iterations are no longer needed.* □

5.4 Convergence Analysis

In this section, convergence results are presented for the proposed primal-dual path-following shifted penalty-barrier method. First, a convergence analysis of the inner iteration Algorithm 5.2 are discussed. Lemma 5.4.1 from Moré and Sorensen [31] is needed to show that the merit function $\mathcal{M}$ is monotonically decreasing.

Lemma 5.4.1. *Suppose $\Phi(\beta) : \mathbb{R} \mapsto \mathbb{R}$ be twice continuously differentiable on the open interval I containing the origin, and assume that $\eta_s \in (0,1)$. Then there must exist an $\bar{\beta} > 0$ such that*

$$\Phi(\beta) \le \Phi(0) + \eta_s \left(\Phi'(0)\beta + \tfrac{1}{2}\Phi'(0)\beta^2 \right),$$

for all $\beta \in [0, \bar{\beta}]$ provided that either $\Phi'(0) < 0$, or $\Phi'(0) = 0$ and $\Phi''(0) < 0$.

Proof. See Lemma 2.2 in Moré and Sorensen [31]. □

Lemma 5.4.2. *Assume that Algorithm 5.2 generates an infinite sequence, i.e., $\nabla\mathcal{M}(v_k) \neq 0$ for all $k \ge 0$, and the sequence $\{x_k\}_{k\ge 0}$ is contained in a compact set. Assume further that $\mathcal{M}$ is bounded below. If either $\nabla\mathcal{M}_k^T \mathit{\Delta} v_k < 0$ or $\nabla\mathcal{M}_k^T \mathit{\Delta} v_k = 0$ and $\mathit{\Delta} v_k^T \nabla^2\mathcal{M}_k \mathit{\Delta} v_k < \eta_1 \mathit{\Delta} v_k^T B_k \mathit{\Delta} v_k$, where $\eta_1 \in (0,1)$, then for any constant $\alpha_k \in (0,1)$, there must exist an $\beta_k \in (0,1)$, such that β_k satisfies the Armijo condition*

$$\mathcal{M}(v_k + \alpha_k\beta_k \mathit{\Delta} v_k) - \mathcal{M}(v_k) \le \eta_s \mathcal{Q}_k^-(\alpha_k\beta_k \mathit{\Delta} v_k),$$

corresponding to Step 16 in Algorithm 5.2. □

Proof. Define $\Phi(\beta) = \mathcal{M}(v_k + \alpha_k\beta \mathit{\Delta} v_k)$, which is twice continuously differentiable. Then the first and second derivative of $\Phi(\beta)$ at zero is given by

$$\Phi'(0) = \alpha_k \nabla \mathcal{M}_k^T \mathit{\Delta} v_k$$

$$\Phi''(0) = \alpha_k^2 \mathit{\Delta} v_k^T \nabla^2 \mathcal{M}_k \mathit{\Delta} v_k,$$

where $\nabla \mathcal{M}_k = \nabla \mathcal{M}(v_k)$ and $\nabla^2 \mathcal{M}_k = \nabla^2 \mathcal{M}(v_k)$. Since $\mathit{\Delta} v_k$ is the solution of the trust region subproblem, which satisfies the positive semidefinite system $(B_k + \sigma_k T_k)\mathit{\Delta} v_k = -\nabla \mathcal{M}_k$. Thus it holds that

$$\Phi'(0) = \alpha_k \nabla \mathcal{M}_k^T \mathit{\Delta} v_k = -\alpha_k \mathit{\Delta} v_k^T (B_k + \sigma_k T_k) \mathit{\Delta} v_k \leq 0.$$

If $B_k + \sigma_k T_k$ is positive definite, then for any $\mathit{\Delta} v_k \neq 0$, $\Phi'(0) < 0$. Otherwise, for positive semidefinite $B_k + \sigma_k T_k$, and $\mathit{\Delta} v_k$ satisfying $\mathit{\Delta} v_k^T (B_k + \sigma_k T_k) \mathit{\Delta} v_k = 0$, it must hold that $\Phi'(0) = 0$, i.e.,

$$\mathit{\Delta} v_k^T B_k \mathit{\Delta} v_k = -\sigma_k \mathit{\Delta} v_k^T T_k \mathit{\Delta} v_k < 0.$$

According to the assumption $\mathit{\Delta} v_k^T \nabla^2 \mathcal{M}_k \mathit{\Delta} v_k < \eta_1 \mathit{\Delta} v_k^T B_k \mathit{\Delta} v_k$, it holds that

$$\Phi''(0) = \alpha_k^2 \mathit{\Delta} v_k^T \nabla^2 \mathcal{M}_k \mathit{\Delta} v_k < \eta_1 \alpha_k^2 \mathit{\Delta} v_k^T B_k \mathit{\Delta} v_k < 0.$$

According to Lemma 5.4.1, it holds that

$$\Phi(\beta) \leq \Phi(0) + \eta_s \left(\Phi'(0)\beta + \tfrac{1}{2}\Phi''(0)\beta^2 \right),$$

which is equivalent to

$$\mathcal{M}(v_k + \alpha_k \beta_k \mathit{\Delta} v_k) - \mathcal{M}(v_k) \leq \eta_s \mathcal{Q}_k^-(\alpha_k \beta_k \mathit{\Delta} v_k),$$

which completes the proof. □

According to Lemma 5.4.1 and Lemma 5.4.2, it has been shown that the sequence of iterates $\{v_k\}_{k\geq 0}$ satisfies $\mathcal{M}(v_{k+1}) < \mathcal{M}(v_k)$ for all k. The next theorem gives the properties of the primal-dual pairs of the iterates generated by Algorithm 5.2, which provides a complete description of the properties of the merit function $\mathcal{M}(x, s, y, w; y^E, w^E, \mu^P, \mu^B)$ for fixed pair

(y^E, w^E, μ^P, μ^B). This proof is similar to the proof of Lemma 3.2 in Gill et al. [20].

Theorem 5.4.3. *Assume that $f(x), c(x)$ are twice continuously differentiable, $\{x_k\}_{k\geq 0}$ are contained in a compact set. Then the iterates $\{v_k\}_{k\geq 0}$ generated by Algorithm 5.2 satisfies the following properties, where (y^E, w^E, μ^P, μ^B) are all fixed,*

- *the sequence $\{s_k\}$, $\{c_k - s_k\}$, $\{y_k\}$, $\{w_k\}$ are uniformly bounded,*
- *for all i, it holds that $\liminf_{k\geq 0}[s_k + \mu^B e]_i > 0$ and $\liminf_{k\geq 0}[w_k]_i > 0$.* □

Proof. To simplify the notation, denote $c_k = c(x_k)$, $f_k = f(x_k)$ and $\mathcal{M}_k = \mathcal{M}(v_k)$. First prove the sequence $\{s_k\}$ are uniformly bounded. Suppose not, since $s_k + \mu^B e > 0$ by construction for all k, then there must exist a subsequence $\mathcal{S}$, and a component i, such that

$$\lim_{k\in\mathcal{S}} [s_k]_i = \infty, \quad \text{and} \quad [s_k]_i \geq [s_k]_j \quad \text{for all} \quad j,$$

Using the assumptions that $\{x_k\}_{k\geq 0}$ are contained in a compact set and f is continuous, it must hold that $\{f_k\}_{k\in\mathcal{S}}$ are bounded below and $\left\{(c_k - s_k)^T y^E\right\}_{k\in\mathcal{S}}$ cannot go to $-\infty$ faster than $\{||s_k||\}_{k\in\mathcal{S}}$. Also $\left\{\frac{1}{2\mu^P}||c_k - s_k||^2\right\}_{k\in\mathcal{S}} \rightarrow +\infty$ in the same rate as $||s_k||^2$, $\{\sum_{i=1}^m [w_k]_i([s_k]_i + \mu^B)\}_{k\in\mathcal{S}}$ are bounded below, and $\left\{\frac{1}{2\mu^P}||c_k - s_k + \mu^P(y_k - y^E)||\right\}_{k\in\mathcal{S}}$ are bounded below by 0. If $-\{\sum_{i=1}^m \mu^B w_i^E \ln([w_k]_i([s_k]_i + \mu^B))\}_{k\in\mathcal{S}}$ is bounded below, then there must exist $\mathcal{S}_1 \subseteq \mathcal{S}$ and a component j satisfying

$$\begin{gathered} \lim_{k\in\mathcal{S}_1} [s_k + \mu^B e]_j [w_k]_j = +\infty, \\ [s_k + \mu^B e]_j [w_k]_j \geq [s_k + \mu^B e]_\ell [w_k]_\ell \quad \text{for all} \quad \ell \quad \text{and} \quad k \in \mathcal{S}_1. \end{gathered} \tag{5.34}$$

These limits imply that $\{\sum_{i=1}^m [w_k]_i([s_k]_i + \mu^B)\}_{k\in\mathcal{S}_1}$ converges to $+\infty$ faster than the sequence $\{-\sum_{i=1}^m \mu^B w_i^E \ln([w_k]_i([s_k]_i + \mu^B))\}_{k\in\mathcal{S}_1}$ converges to $-\infty$. Then it must follow that $\lim_{k\in\mathcal{S}} \mathcal{M}_k = +\infty$, which contradicts Lemma 5.4.2. It follows that the sequence $\{s_k\}$ is uniformly bounded.

It can also be shown by contradiction that $\{y_k\}$ are uniformly bounded. If this were not the case, then there must exist a subsequence $\mathcal{S}$ and a component i such that

$$\lim_{k\in\mathcal{S}} |[y_k]_i| = +\infty, \quad \text{and} \quad |[y_k]_i| \geq |[y_k]_j| \quad \text{for all} \;\; j,$$

For $k \in \mathcal{S}$, the sequence $\{f_k\}$, $\left\{(c_k - s_k)^T y^E\right\}$, $\left\{\frac{1}{2\mu^P}||c_k - s_k||^2\right\}$ are uniformly bounded, $\left\{\frac{1}{2\mu^P}||c_k - s_k + \mu^P(y_k - y^E)||^2\right\}_{k\in\mathcal{S}}$ converges to $+\infty$ at the rate of $[y_k]_i^2$ since $\{s_k\}$ is uniformly bounded. The sequence $\{-\sum_{i=1}^m \mu^B w_i^E \ln([s_k]_i + \mu^B)\}_{k\in\mathcal{S}}$ is uniformly bounded below because $w^E > 0$, and $\{\sum_{i=1}^m [w_k]_i([s_k]_i + \mu^B)\}_{k\in\mathcal{S}}$ is bounded below by zero trivially, thus if $\{-\sum_{i=1}^m \mu^B w_i^E \ln([w_k]_i([s_k]_i + \mu^B))\}_{k\in\mathcal{S}}$ is bounded below, then it follows that $\lim_{k\in\mathcal{S}} \mathcal{M}_k = +\infty$, which contradicts Lemma 5.4.2. Thus $\{-\sum_{i=1}^m \mu^B w_i^E \ln([w_k]_i([s_k]_i + \mu^B))\}_{k\in\mathcal{S}}$ must converge to $-\infty$, so condition (5.34) is satisfied for some subsequence $\mathcal{S}_1 \subseteq \mathcal{S}$ and a component j. Then for $k \in \mathcal{S}_1$ and this j, it holds that $\{\sum_{i=1}^m [w_k]_i([s_k]_i + \mu^B)\}_{k\in\mathcal{S}_1}$ converges to $+\infty$ faster than the sequence $\{-\sum_{i=1}^m \mu^B w_i^E \ln([w_k]_i([s_k]_i + \mu^B))\}_{k\in\mathcal{S}_1}$ converging to $-\infty$, which contradicts Lemma 5.4.2 again. So $\{y_k\}$ must be uniformly bounded.

Next we show that $\{w_k\}$ is uniformly bounded. If not there must exist a subsequence $\mathcal{S}$ and a component i such that

$$\lim_{k\in\mathcal{S}} [w_k]_i = +\infty, \quad \text{and} \quad [w_k]_i \geq [w_k]_j \quad \text{for all} \;\; j, \tag{5.35}$$

It follows that there must exist $\mathcal{S}_1 \subseteq \mathcal{S}$ and $\mathcal{J} \subseteq \{1, 2, \ldots, m\}$ satisfying the condition

$$\lim_{k\in\mathcal{S}_1} [w_k]_j = +\infty \;\; \text{for all} \;\; j \in \mathcal{J}, \quad \text{and} \quad \{[w_k]_j : j \notin \mathcal{J}, \;\; k \in \mathcal{S}_1\} \;\; \text{is bounded.}$$

As $\{s_k\}$ and $\{y_k\}$ are uniformly bounded, then for $k \in \mathcal{S}$, the sequence $\{f_k\}$, $\left\{(c_k - s_k)^T y^E\right\}$, $\left\{\frac{1}{2\mu^P}||c_k - s_k||^2\right\}$, and $\left\{\frac{1}{2\mu^P}||c_k - s_k + \mu^P(y_k - y^E)||^2\right\}$ are bounded. Then from condition (5.35), it follows that $\{-\sum_{i=1}^m \mu^B w_i^E \ln(([s_k]_i + \mu^B)^2[w_k]_i)\}_{k\in\mathcal{S}}$ is asymptotically bounded below by $\{-\sum_{i=1}^m \mu^B w_i^E \ln([w_k]_i)\}_{k\in\mathcal{S}}$. Thus it must hold that

$$[w_k]_j([s_k]_j + \mu^B) = O(\ln([w_k]_i)) \;\; \text{for all} \;\; j \in \{1, 2, \ldots, m\}, \tag{5.36}$$

otherwise $\lim_{k\in\mathcal{S}} \mathcal{M}_k = +\infty$, which contradicts Lemma 5.4.2. It follows from (5.36) that there must exist a constant $\kappa_1 > 0$ such that

$$\begin{aligned}\ln([s_k]_j + \mu^{B}) &\leq \ln\left(\frac{\kappa_1 \ln([w_k]_i)}{[w_k]_j}\right) \\ &= \ln(\kappa_1) + \ln(\ln([w_k]_i)) - \ln([w_k]_j),\end{aligned}$$

for all $1 \leq j \leq m$ and k sufficient large. Define $\alpha = [w^{E}]_i/4||w^{E}||_1 > 0$, which is well-defined because $w^{E} > 0$ by construction. Then it holds that

$$2\ln(\ln([w_k]_i)) - \ln([w_k]_j) \leq \alpha \ln([w_k]_i), \tag{5.37}$$

for all $j \in \mathcal{J}$ and sufficient large $k \in \mathcal{S}_1$. Then there exist positive constants κ_2, κ_3 satisfying

$$\begin{aligned}-\sum_{i=1}^{m} \mu^{B} w_i^{E} \ln\big(([s_k]_i + \mu^{B})^2 [w_k]_i\big) &\geq -\kappa_2 - \mu^{B} \sum_{j\in\mathcal{J}} w_j^{E} \left(2\ln([s_k]_j + \mu^{B}) + \ln([w_k]_j)\right) \\ &\geq -\kappa_2 - \mu^{B} \sum_{j\in\mathcal{J}} w_j^{E} \left(2\ln(\kappa_1) + 2\ln(\ln([w_k]_i)) - \ln([w_k]_j)\right) \\ &\geq -\kappa_3 - \mu^{B} \sum_{j\in\mathcal{J}} w_j^{E} \left(2\ln(\ln([w_k]_i)) - \ln([w_k]_j)\right) \\ &\geq -\kappa_3 - \mu^{B} w_i^{E} \left(2\ln(\ln([w_k]_i)) - \ln([w_k]_i)\right).\end{aligned}$$

Combining these values with condition (5.37) gives

$$\begin{aligned}&-\sum_{i=1}^{m} \mu^{B} w_i^{E} \ln\big(([s_k]_i + \mu^{B})^2 [w_k]_i\big) \\ &\quad\geq -\kappa_3 - \mu^{B} w_i^{E} \left(2\ln(\ln([w_k]_i)) - \ln([w_k]_i)\right) - \mu^{B} \sum_{j\in\mathcal{J}, j\neq i} w_j^{E} \left(2\ln(\ln([w_k]_i)) - \ln([w_k]_j)\right) \\ &\quad\geq -\kappa_3 - \mu^{B} w_i^{E} \left(2\ln(\ln([w_k]_i)) - \ln([w_k]_i)\right) - \mu^{B} \sum_{j\in\mathcal{J}, j\neq i} w_j^{E} \alpha \ln([w_k]_i) \\ &\quad\geq -\kappa_3 - \mu^{B} w_i^{E} \left(2\ln(\ln([w_k]_i)) - \ln([w_k]_i)\right) - \mu^{B} \alpha \ln([w_k]_i) ||w^{E}||,\end{aligned} \tag{5.38}$$

holds for sufficiently large $k \in \mathcal{S}_1$. Condition (5.38) together with condition (5.37) gives

$$-\sum_{i=1}^{m} \mu^{B} w_i^{E} \ln\big(([s_k]_i + \mu^{B})^2 [w_k]_i\big) \geq -\kappa_3 + \frac{1}{4}\mu^{B} \ln([w_k]_i), \tag{5.39}$$

which implies that

$$\lim_{k \in \mathcal{S}_1} \left(-\sum_{i=1}^{m} \mu^{B} w_i^{E} \ln\big(([s_k]_i + \mu^{B})^2 [w_k]_i\big) \right) = +\infty. \tag{5.40}$$

It follows immediately that $\lim_{k \in \mathcal{S}_1} \mathcal{M}_k = +\infty$, which contradicts Lemma 5.4.2. Thus it must hold that $\{w_k\}$ is bounded.

Based on the results of part 1, part 2 can also be established by contradiction in a similar argument. As before, for all $k \in \mathcal{S}$, the sequence $\{f_k\}$, $\big\{(c_k - s_k)^T y^{E}\big\}$, $\Big\{\frac{1}{2\mu^{P}}||c_k - s_k||^2\Big\}$, $\Big\{\frac{1}{2\mu^{P}}||c_k - s_k + \mu^{P}(y_k - y^{E})||^2\Big\}$, and $\{\sum_{i=1}^{m}[w_k]_i([s_k]_i + \mu^{B})\}$ are all bounded. If it holds that $\liminf_{k \in \mathcal{S}}[s_k + \mu^{B} e]_i = 0$ for some subset $\mathcal{S}$ and some component i, then condition (5.40) holds, so $\lim_{k \in \mathcal{S}_1} \mathcal{M}_k = +\infty$, which contradicts to Lemma 5.4.2. Thus $\liminf_{k \geq 0}[s_k + \mu^{B} e]_i > 0$ for all i. Similarly, it holds that $\liminf_{k \geq 0}[w_k]_i > 0$ for all i. □

Based on Theorem 5.4.3, we can further prove the following Theorem 5.4.4, which is used for the proof in the main convergence result of Theorem 5.4.5.

Theorem 5.4.4. *Assume all the assumption in Theorem 5.4.3 holds, then the iterates $\{v_k\}_{k \geq 0}$ generated by Algorithm 5.2 further satisfies the following properties, where the pair $(y^{E}, w^{E}, \mu^{P}, \mu^{B})$ are all fixed,*

- *$\{\pi_k^{P}\}$, $\{\pi_k^{B}\}$, and $\{\nabla \mathcal{M}(v_k)\}$ are uniformly bounded,*
- *there must exist a scalar $\mathcal{M}_{low}$ satisfying $\mathcal{M}(v_k) \geq \mathcal{M}_{low} > -\infty$ for all k.* □

Proof. To show part 1, by the definition of $\pi_k^{P} = \pi^{P}(x_k, s_k) = y^{E} - (c_k - s_k)/\mu^{P}$, it is immediate that $\{\pi_k^{P}\}$ are uniformly bounded since $\{c_k - s_k\}$ are uniformly bounded and y^{E}, μ^{P} are fixed. Similarly, $\pi_k^{B} = \pi^{B}(s_k) = \mu^{B}(S_k + \mu^{B} I)^{-1} w^{E}$, since $\liminf_{k \geq 0}[s_k + \mu^{B} e]_i > 0$ for all i, $\{s_k\}$ are uniformly bounded, and w^{E}, μ^{B} are fixed, $\big\{\pi_k^{B}\big\}$ are uniformly bounded.

Finally, the uniform boundedness of $\{\nabla\mathcal{M}(v_k)\}$ immediately follows based on the uniform boundedness of $\{s_k\}$, $\{c_k - s_k\}$, $\{y_k\}$, $\{w_k\}$, $\{\pi_k^P\}$, $\{\pi_k^B\}$ from its definition in (5.22).

For part 2, it can be shown that each term in the merit function $\mathcal{M}$ are bounded below. $\{f_k\}$ are uniformly bounded below by the continuity of $f(x)$ and $\{x_k\}$ are contained in compact set. $\left\{(c_k - s_k)^T y^E\right\}$ are bounded since $\{c_k - s_k\}$ are uniformly bounded and y^E are fixed. The two penalty terms $\left\{\frac{1}{2\mu^P}||c_k - s_k||^2\right\}$, and $\left\{\frac{1}{2\mu^P}||c_k - s_k + \mu^P(y_k - y^E)||^2\right\}$ are nonnegative, trivially bounded below by 0. $\{-\sum_{i=1}^m \mu^B w_i^E \ln(([s_k]_i + \mu^B)^2[w_k]_i)\}$ is uniformly bounded below since $\liminf_{k\geq 0}[s_k + \mu^B e]_i > 0$ for all i, $\{w_k\}$ are uniformly bounded and μ^B, w^E are fixed. The last part $\{\sum_{i=1}^m [w_k]_i([s_k]_i + \mu^B)\}$ are nonnegative since $w_k > 0$ by construction for all k. □

Based on Theorem 5.4.3 and Theorem 5.4.4, the next theorem provides the main convergence result of inner iteration.

Theorem 5.4.5. *Suppose all the assumptions in Theorem 5.4.3 hold, then the sequence* $\{v_k\}_{k\geq 0}$ *generated by Algorithm 5.2 satisfies* $\lim_{k\to\infty}\nabla\mathcal{M}(v_k) = 0$. □

Proof. The proof is by contradiction. Denote $\mathcal{M}(v_k)$ to be $\mathcal{M}_k$. Suppose there exists a constant $\varepsilon > 0$ and a subsequence $\mathcal{S}$ such that $||\nabla\mathcal{M}_k|| \geq \varepsilon$ for all $k \in \mathcal{S}$. Then it follows from Theorem 5.4.3 and Theorem 5.4.4 that $\lim_{k\in\mathcal{S}}\mathcal{M}_k = \mathcal{M}_{\min} > -\infty$. Then by Lemma 5.4.2, the Armijo condition (Step 16, Algorithm 5.2) is satisfied for all $k \in \mathcal{S}$. Then it must hold that

$$\lim_{k\to\infty} \alpha_k \nabla\mathcal{M}_k^T \varDelta v_k = 0, \tag{5.41}$$

which further implies that $\lim_{k\in\mathcal{S}}\alpha_k = 0$. This implies that there must exist a subsequence $\mathcal{S}_1 \subseteq \mathcal{S}$ and a component i, such that for all $k \in \mathcal{S}$, *either*

$$[s_k + \alpha_k \varDelta s_k]_i + \mu^B > 0, \;\text{ and }\; [s_k + \frac{1}{\gamma_s}\alpha_k \varDelta s_k]_i + \mu^B \leq 0, \tag{5.42}$$

where $\gamma_s \in (0,1)$ is the line-search factor in Algorithm 5.2, *or*

$$[w_k + \alpha_k \varDelta w_k]_i > 0, \ \text{ and } \ [w_k + \frac{1}{\gamma_s}\alpha_k \varDelta w_k]_i \leq 0. \tag{5.43}$$

Without loss of generality, suppose condition (5.43) holds. Then there must exist some $\varepsilon > 0$ such that

$$\varepsilon < w_{k+1} = w_k + \alpha_k \varDelta w_k = w_k + \frac{1}{\gamma_s}\alpha_k \varDelta w_k - \frac{1}{\gamma_s}\alpha_k \varDelta w_k + \alpha_k \varDelta w_k, \tag{5.44}$$

for some sufficiently large k. So it must hold that

$$w_k + \frac{1}{\gamma_s}\alpha_k \varDelta w_k > \varepsilon + \frac{1}{\gamma_s}\alpha_k \varDelta w_k - \alpha_k \varDelta w_k = \varepsilon + \alpha_k \varDelta w_k (\frac{1}{\gamma_s} - 1) > 0, \tag{5.45}$$

which contradicts to condition (5.43), thus it must hold that $\lim_{k\to\infty} \nabla \mathcal{M}(v_k) = 0$. □

So far the convergence analysis of the inner iterations has been established, but the adjustment of the pair $(y^{E}, w^{E}, \mu^{P}, \mu^{B})$ must be done carefully in the outer iteration to enforce a strong convergence property. To distinguish between the outer and inner iterations, it is convenient to define

$$\mathcal{O} = \{k : \text{iteration } k \text{ is an outer iteration (O-iteration)}\}$$
$$\mathcal{F} = \{k : \text{iteration } k \text{ is an inner iteration (F-iteration)}\}.$$

First, Theorem 5.4.6 shows that any limit point of an infinite sequence of $\mathcal{O}$ iterations must be CAKKT point (see Definition 2.1.11).

Theorem 5.4.6. *Suppose all the assumptions in Theorem 5.4.3 hold. If $|\mathcal{O}| = \infty$, then there exists a limit point (x^*, s^*) of the sequence $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{O}}$ and any such point must be a CAKKT point.* □

Proof. By the assumption that $\{x_k\}_{k\in\mathcal{O}}$ are contained in a compact set, then there must exist a subsequence $\mathcal{K} \subseteq \mathcal{O}$ and one limit point of $\{x_k\}_{k\in\mathcal{K}}$, denoted by x^*. By the continuity of $c(x)$, it follows that $\lim_{k\in\mathcal{K}} c(x_k) = c(x^*)$. As $|\mathcal{O}| = \infty$, according to Step 7 of Algorithm 5.1, it must hold that $\lim_{k\in\mathcal{K}} ||F_k|| = 0$. Then it follows that $\lim_{k\in\mathcal{K}}(c(x_k) - s_k) = 0$ since $y_k^E = y_k$ is updated infinitely many often. Define $s^* = c(x^*)$, it follows that $\{s_k\}_{k\in\mathcal{K}} \rightarrow \lim_{k\in\mathcal{K}} c(x_k) = c(x^*) = s^*$, which implies that $c(x^*) - s^* = 0$. For all $i \in \{1, 2, \ldots, m\}$, define the following two sets:

$$\mathcal{Q}_1 = \{k : [q_1(v_k)]_i \le [q_2(v_k, \mu_k^B)]_i\} \; \mathcal{Q}_2 = \{k : [q_2(v_k, \mu_k^B)]_i < [q_1(v_k)]_i\},$$

where the definition of $q_1(v_k)$ and $q_2(v_k, \mu_k^B)$ can be referred in equation (5.14). If $|\mathcal{K} \cap \mathcal{Q}_1| = \infty$, then it follows from $\{\kappa_{comp}(v_k, \mu_k^B)\}_{k\in\mathcal{K}} \rightarrow 0$ that $[s^*]_i = \lim_{k\in\mathcal{K}\cap\mathcal{Q}_1}[s_k]_i \ge 0$. Similarly, if $|\mathcal{K} \cap \mathcal{Q}_2| = \infty$, then

$$[s^*]_i = \lim_{k\in\mathcal{K}\cap\mathcal{Q}_2}[s_k]_i = \lim_{k\in\mathcal{K}\cap\mathcal{Q}_2}[s_k + \mu_k^B e]_i \ge 0,$$

from the definition of $\mathcal{Q}_2$. Then it is immediate that $[s^*]_i \ge 0$ as claimed. Thus (x^*, s^*) must be feasible.

It remains to show that (x^*, s^*) must be a CAKKT point. Consider the sequence $\{x_k, \bar{s}_k, y_k, w_k\}_{k\in\mathcal{K}}$ where $\bar{s}_k = s_k$ if $k \in \mathcal{Q}_1$ and $\bar{s}_k = s_k + \mu_k^B e$ if $k \in \mathcal{Q}_2$. If $|\mathcal{O} \cap \mathcal{Q}_2| < \infty$, since $\lim_{k\in\mathcal{K}} s_k = s^*$, it follows that $\lim_{k\in\mathcal{K}} \{[\bar{s}_k]_i\}_{k\in\mathcal{K}} \rightarrow [s^*]_i$. However, if $|\mathcal{O} \cap \mathcal{Q}_2| = \infty$, since $\lim_{k\in\mathcal{K}} \kappa_{comp}(v_k, \mu_k^B) = 0$, then by the definition of $\mathcal{Q}_2$, it is immediate that $\{\mu_k^B\} \rightarrow 0$, giving $\lim_{k\in\mathcal{K}} \{[\bar{s}_k]_i\}_{k\in\mathcal{K}} \rightarrow [s^*]_i$ for all $i \in \{1, 2, \ldots, m\}$. Thus $\lim_{k\in\mathcal{K}} \bar{s}_k = s^*$.

Then it can be shown that $\{x_k, \bar{s}_k, y_k, w_k\}_{k\in\mathcal{K}}$ satisfies the definition of CAKKT point defined in (2.1). It follows from the previous argument that $\lim_{k\in\mathcal{K}} ||F(v_k)|| = 0$, which implies that $\lim_{k\in\mathcal{K}}(g_k - J_k^T y_k) = 0$ and $\lim_{k\in\mathcal{K}}(y_k - w_k) = 0$. Step 4 in Algorithm 5.2 implies that $w_k \ge 0$ for all k, thus $\lim_{k\in\mathcal{K}} w_k \ge 0$. Finally, if $|\mathcal{K} \cap \mathcal{Q}_1| = \infty$, then the definition of $\bar{s}_k$, $q_i(v_k)$ and $\lim_{k\in\mathcal{K}} \kappa_{comp}(v_k, \mu_k^B) = 0$ imply that $\lim_{k\in\mathcal{K}\cap\mathcal{Q}_1}[w_k \cdot \bar{s}_k]_i = 0$ for all i. A similar argument yields that $\lim_{k\in\mathcal{K}\cap\mathcal{Q}_2}[w_k \cdot \bar{s}_k]_i = 0$. Thus these two cases lead to the conclusion that $\lim_{k\in\mathcal{K}} w_k \cdot \bar{s}_k = 0$, which implies that (x^*, s^*) must be a CAKKT point. □

Then the next theorems will show that if $|\mathcal{O}| < \infty$, then all limit points of the set of $\mathcal{F}$-iterations are *infeasible stationary points*, which is defined below.

Definition 5.4.1 (Infeasible stationary point)**.** *The pair (x^*, s^*) is called an infeasible stationary point if $c(x^*) - s^* \neq 0$ and the following optimality conditions hold*

$$J(x^*)^T(c(x^*) - s^*) = 0, \quad s^* \geq 0,$$
$$s^* \cdot (c(x^*) - s^*) = 0, \quad c(x^*) - s^* \leq 0,$$

where $J(x^*)$ represents the Jacobi of $c(x)$ at x^*. □

Theorem 5.4.7. *Suppose all the assumptions in Theorem 5.4.3 hold. If $|\mathcal{O}| < \infty$, then there must exist at least one limit point (x^*, s^*) of $\{(x_k, s_k)\}_{k \in \mathcal{F}}$, and any such limit point is an infeasible stationary point defined in Definition 5.4.1.* □

Proof. First, it is shown that $\{s_k\}_{k \in \mathcal{F}}$ is bounded. If this were not the case, then there must exist a subsequence $\mathcal{K} \subseteq \mathcal{F}$ and a component i for which $\lim_{k \in \mathcal{K}} [s_k]_i = +\infty$. Since $\{w_k^E\}_{k \in \mathcal{K}}$ are bounded and $\{\mu_k^B\}_{k \in \mathcal{K}} > 0$ are monotonically decreasing, it follows that $\lim_{k \in \mathcal{K}} [\pi_k^B]_i = 0$ by the definition of π^B in equation (5.16). A similar argument gives that $\lim_{k \in \mathcal{K}} [\pi_k^P]_i = 0$. However, the boundedness of $\{y_k^E\}$ and the assumption that $\lim_{k \in \mathcal{K}} [s_k]_i = +\infty$ imply that $\lim_{k \in \mathcal{K}} [c(x_k)]_i = +\infty$ which is impossible since $\{x_k\}$ are assumed to be in a compact set and $c(x)$ is continuous. Thus $\{s_k\}_{k \in \mathcal{K}}$ must be bounded.

The boundedness of $\{s_k\}_{k \in \mathcal{K}}$ and the fact that $\{x_k\}$ are contained in a compact set imply that there must exist a subsequence $\mathcal{K} \subseteq \mathcal{F}$ and a limit point (x^*, s^*), such that $\lim_{k \in \mathcal{K}} (x_k, s_k) = (x^*, s^*)$. It remains to show that (x^*, s^*) satisfies the conditions in Definition 5.4.1. Since $\{y_k^E\}$ are bounded and $\lim_{k \in \mathcal{F}} |y_k - w_k| = 0$ if $|\mathcal{F}| = \infty$, then as $\{\mu_k^P\} \to 0$, Lemma 5.3.5 gives that for any $k \in \mathcal{K}$, it holds that

$$c(x_{k+1}) - s_{k+1} \leq \mu_k^P \left(y_k^E + \tfrac{1}{2}(w_{k+1} - y_{k+1}) \right) \to 0,$$

which implies that $c(x^*) - s^* \leq 0$. If $|\mathcal{F}| = \infty$ and step 7 in Algorithm 5.1 is satisfied infinitely many times, then the update $\mu_{k+1}^B = \frac{1}{2}\mu_k^B$ forces $\left\{\mu_{k+1}^B\right\} \to 0$. Since $s_k + \mu_k^B e > 0$ by construction, thus $s^* \geq 0$ as claimed.

For a proof of $J(x^*)^T(c(x^*) - s^*) = 0$, notice that the gradient of the merit function

$\mathcal{M}$ must satisfy

$$\lim_{k\in\mathcal{K}} \nabla_x \mathcal{M}(v_{k+1}; y_k^E, w_k^E, \mu_k^P, \mu_k^B) = 0.$$

As $|\mathcal{F}| = \infty$ and according to Lemma 5.3.1, multiply $\nabla_x \mathcal{M}(v_{k+1}; y_k^E, w_k^E, \mu_k^P, \mu_k^B)$ by μ_k^P gives that

$$\lim_{k\in\mathcal{K}} \left(\mu_k^P g(x_{k+1}) - J(x_{k+1})^T \left(\mu_k^P \pi_{k+1}^P + \mu_k^P(\pi_{k+1}^P - y_{k+1})\right)\right) = 0,$$

because $\lim_{k\in\mathcal{K}} x_k = x^*$ and $\lim_{k\in\mathcal{K}} \mu_k^P = 0$. Then it holds that

$$\lim_{k\in\mathcal{K}} \left(-J(x_{k+1})^T(\mu_k^P \pi_{k+1}^P)\right) = \lim_{k\in\mathcal{K}} \left(-J(x_{k+1})^T \left(\mu_k^P y_k^E - c(x_{k+1}) + s_{k+1}\right)\right) = 0.$$

This limit, together with the fact that the sequence $\{y_k^E\}$ is bounded, $\{\mu_k^P\} \to 0$, and $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{K}} \to (x^*, s^*)$ imply that $J(x^*)^T(c(x^*) - s^*) = 0$.

Finally, it remains to show that $s^* \cdot (c(x^*) - s^*) = 0$. From the preceding argument, it holds that $\lim_{k\in\mathcal{K}}(\pi_{k+1}^B - \pi_{k+1}^P) = 0$. The boundedness of the sequence $\{\mu_k^P(s_{k+1} + \mu_k^B I)\}_{k\in\mathcal{K}}$ yields the following relation

$$\lim_{k\in\mathcal{K}} (\mu_k^B \mu_k^P w_k^E - \mu_k^P(S_{k+1} + \mu_k^B I) y_k^E + (S_{k+1} + \mu_k^B I)(c(x_{k+1}) - s_{k+1})) = 0.$$

Moreover, as $\lim_{k\in\mathcal{K}} s_k = s^*$, $\{\mu_k^P\} \to 0$ and $\{y_k^E\}$, $\{w_k^E\}$ are bounded. It holds that

$$\lim_{k\in\mathcal{K}} (S_{k+1} + \mu_k^B I)(c(x_{k+1}) - s_{k+1}) = 0. \tag{5.46}$$

Because $c(x^*) - s^* \neq 0$, there must exist a component i, such that $[c(x^*) - s^*]_i \neq 0$. Combining this with the fact that $\lim_{k\in\mathcal{K}}(x_k, s_k) = (x^*, s^*)$ shows that $\lim_{k\in\mathcal{K}}[s_{k+1}]_i + \mu_k^B = 0$. Since $s^* \geq 0$, it must hold that $\{\mu_k^B\}_{k\in\mathcal{K}} \to 0$. As $\{\mu_k^B\}$ is also monotonically decreasing to 0, then it is immediate from relation (5.46) that $s^*(c(x^*) - s^*) = 0$ which completes the proof. □

At the end of this section, Theorem 5.4.8 provides the main convergence results for the outer iteration.

Theorem 5.4.8. *Suppose that the assumptions of Theorem 5.4.3 hold. Then one of the following two results must hold:*

- $|\mathcal{O}| = \infty$, *then the limit point of* $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{O}}$ *exists, denoted by* (x^*, s^*)*, which is a CAKKT point of problem (NIPs). If the Cone-Continuity Property (CCP) holds at* (x^*, s^*)*, then* (x^*, s^*) *is a KKT point point for problem (NIPs).*

- $|\mathcal{O}| < \infty$, $|\mathcal{F}| = \infty$*. Then the limit point of* $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{F}}$ *exists and every such limit point* (x^*, s^*) *is an infeasible stationary point.* □

Proof. It is clear one of the two cases must occur. If $|\mathcal{O}| = \infty$, then according to Theorem 5.4.6, the conclusion immediately follows. The proof of CAKKT point under CCP is in fact a KKT point can be found in Andreani et al. [1]. Part 2 follows immediately from Theorem 5.4.7. □

5.5 Acknowledgement

This chapter, in part is currently being prepared for submission for publication of the material. Su, Fangyao; Gill, Philip E. The dissertation author was the primary investigator and author of this material.

Chapter 6

Numerical Implementations

In this chapter, numerical results of the proposed primal-dual path-following augmented Lagrangian method for (NEP) in Chapter 3, and the proposed primal-dual path-following shifted penalty-barrier method for (NIP) in Chapter 5 are presented. In optimization society, CUTEst test collection is commonly used to provide a fair comparison of the optimization algorithms. It stands for Constrained and Unconstrained Testing Environment, previously called CUTEr, which contains a set of test problems for both linear and nonlinear, constrained and unconstrained optimization problems (see Gould, Orban and Toint [28, 29]). The implementation of the proposed algorithms was done in MATLAB R2015ASP1 and run on the 2014 MacbookPro with 8 GB 1600 MHz processor.

Although each problem in CUTEst has its own individual form, the most general format can be described as

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}} && f(x) \\ &\text{subject to} && c_\ell \le c(x) \le c_u, \quad a_\ell \le Ax \le a_u, \quad x_\ell \le x \le x_u, \end{aligned}$$

where $f(x)$ is the objective function, $A \in \mathbb{R}^{m\times n}$, and $c_\ell \le c_u$, $a_\ell \le a_u$, $x_\ell \le x_u$ are constant vectors that represent the lower and upper bounds of nonlinear constraints, linear constraints and the primal variables respectively.

Section 6.1 concerns the primal-dual path-following augmented Lagrangian method proposed in Chapter 3. The numerical results of the primal-dual path-following shifted penalty-barrier method proposed in Chapter 5 are given in Section 6.2. Both conventional trust-region method and a combined trust-region line-search method have been applied for minimizing the merit function and their performance are compared. It has been shown that the combined trust-region line-search method generally requires fewer matrix factorizations and fewer iterations compared with the conventional trust-region method, which brings numerical benefits for minimizing the shifted penalty-barrier function.

Table 6.1 defines the headings and their meanings for all numerical tables in this chapter.

Table 6.1: Definition of the Headings

Heading	Meaning
prob	Problem name in CUTEst
m	Number of constraints
n	Number of primal variables
Opt-f(x)	Optimal value of objective function
Opt-First	Norm of first order optimality
Opt-Stny	Norm of stationarity, i.e., $\|\|g(x^*) - J(x^*)^T y^*\|\|$
$\|\|c(x^*)\|\|$	Norm of constraint violations
OItr	Number of outer iterations
FItr	Number of inner iterations
CPU(s)	Required CPU time (in seconds)
Conv	Exit code:
	1: Optimal solution found
	0: Failure due to iteration limit reached
	2: Converge to an infeasible stationary point
	3: Failure due to time limit reached (10800 seconds)

6.1 Numerical Results of Primal-Dual Path-Following Augmented Lagrangian Method

This section provides the numerical results of the primal-dual path-following augmented Lagrangian method (PDAL) described in Chapter 3. Problems from CUTEst in all-equality form are tested, i.e., in the form (NEP):

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

The proposed algorithm has both "outer" and "inner" iterations, defined by "OItr" and "FItr" in the table. In the outer iteration, the penalty parameter and multipliers estimate are updated, and the optimality conditions are tested. In the inner iteration, the merit function $\mathcal{M}(x, y; y_k^E, \mu)$ defined in (3.17) is minimized, and the penalty parameter is allowed to increase sometimes for numerical benefits.

The default parameters are listed in Table 6.2. The specific meaning of each parameter can be referred to Algorithm 3.3 and Algorithm 3.4 in Chapter 3.

Table 6.2: Default parameters used in PDAL

Parameter	value	Parameter	value
a_s	0.1	ε_s	1e-7
σ_s	0.2	γ_c	0.5
ρ_s	0.9	α_0	1.0
η_s	1.0e-2	μ_0	0.1
x_0	prob.x	MaxIItn	10
Inf	1e+15	MaxItn	1500

All together, 73 out of the 74 tested problems are solved successfully with the optimal solution found. The only failure is MSS2, which most existing software finds hard to solve.

Table 6.3: Numerical Results of PDAL

Prob	m	n	Opt-f(x)	Opt-Stny	$\|\|c(x^*)\|\|$	OItr	FItr	CPU(s)	Conv
ARGTRIG	200	200	0.00000e+00	5.49084e-10	1.41947e-10	5	0	0.229810	1
BT1	1	2	-1.00000e+00	2.84217e-14	1.95399e-14	9	14	0.324616	1
BT2	1	3	3.25682e-02	2.75224e-09	1.46824e-09	12	0	0.067248	1
BT3	3	5	4.09302e+00	8.88178e-16	6.75581e-08	5	0	0.042930	1
BT4	2	3	-4.55105e+01	6.99231e-09	4.80356e-15	9	11	0.476788	1
BT5	2	3	9.61715e+02	5.88132e-08	1.58882e-14	10	10	0.247276	1
BT6	2	5	2.77045e-01	3.41272e-08	1.00357e-13	12	3	0.057052	1
BT7	3	5	3.06499e+02	7.82834e-08	8.88178e-16	42	118	0.435378	1
BT8	2	5	1.00000e+00	9.13233e-08	9.97801e-08	13	0	0.011517	1
BT9	2	4	-1.00000e+00	1.46397e-08	2.43629e-10	10	0	0.008416	1
BT10	2	2	-1.00000e+00	1.17402e-11	7.23266e-15	8	0	0.011576	1
BT11	3	5	8.24891e-01	4.89242e-09	1.58955e-11	7	1	0.013968	1
BT12	3	5	6.18812e+00	7.86596e-11	1.52906e-12	6	0	0.006814	1
BOOTH	2	2	0.00000e+00	1.22277e-08	1.56796e-09	4	0	0.067839	1
BROWNALE	200	200	0.00000e+00	4.00449e-09	3.83438e-11	7	0	0.283986	1
BYRDSPHR	2	3	-4.68330e+00	8.03247e-09	2.17039e-08	7	20	0.348069	1
CHANDHEQ	100	100	0.00000e+00	8.83994e-08	1.18458e-08	15	0	0.283138	1
CHANDHEU	500	500	0.00000e+00	8.99585e-08	2.65607e-08	15	0	6.880703	1
CLUSTER	2	2	0.00000e+00	3.33714e-12	1.52494e-11	9	0	0.010396	1
CUBENE	2	2	0.00000e+00	8.30210e-09	2.54709e-10	5	1	0.079753	1
EIGMAXB	101	101	-9.67435e-04	5.00775e-10	2.12520e-15	13	60	0.717963	1
EIGMAXC	202	202	-1.00000e+00	1.09871e-08	6.63260e-15	9	10	0.863621	1
EIGMINB	101	101	9.67435e-04	1.01473e-11	1.83735e-15	14	63	0.732214	1
EIGMINC	202	202	1.00000e+00	9.21129e-09	5.04252e-14	9	10	0.805016	1
GENHS28	8	10	9.27174e-01	2.22045e-16	3.68100e-08	4	0	0.012480	1
GOTTFR	2	2	0.00000e+00	6.26224e-13	7.39327e-14	8	11	0.167976	1
HATFLDF	3	3	0.00000e+00	3.00428e-13	8.21021e-17	11	20	0.223787	1
HATFLDG	25	25	0.00000e+00	1.87041e-10	4.81917e-12	7	10	0.123092	1
HEART6	6	6	0.00000e+00	3.42552e-18	1.23200e-14	40	260	6.615571	1
HEART8	8	8	0.00000e+00	9.05655e-15	3.33067e-16	12	30	0.181248	1
HIMMELBA	2	2	0.00000e+00	2.97834e-08	1.03777e-08	4	0	0.066296	1
HIMMELBC	2	2	0.00000e+00	8.08462e-08	2.41950e-08	6	10	0.135004	1
HIMMELBE	3	3	0.00000e+00	8.23424e-10	1.81998e-10	5	0	0.066122	1
HS100LNP	2	7	6.80630e+02	1.77636e-15	2.84556e-14	20	10	0.098372	1
HS111LNP	3	10	-4.77611e+01	6.10558e-08	3.05325e-08	10	5	0.048769	1
HS26	1	3	2.81121e-12	1.72766e-08	2.35156e-08	20	12	0.022988	1
HS27	1	3	4.00000e-02	1.08568e-12	5.58442e-14	6	1	0.083247	1
HS28	1	3	0.00000e+00	4.43728e-16	4.44089e-16	3	0	0.007645	1

Table 6.3: Numerical Results of PDAL (continued)

Prob	m	n	Opt-f(x)	Opt-Stny	$\|\|c(x^*)\|\|$	OItr	FItr	CPU(s)	Conv
HS39	2	4	-1.00000e+00	1.46397e-08	2.43629e-10	10	0	0.037938	1
HS40	3	4	-2.50000e-01	5.93110e-09	1.34016e-11	7	2	0.054932	1
HS42	2	4	1.38579e+01	1.66859e-09	9.03895e-10	6	0	0.066704	1
HS46	2	5	6.19255e-10	8.54932e-08	1.39065e-09	18	20	1.093597	1
HS47	3	5	1.42212e-12	3.78873e-08	1.33944e-08	19	0	0.034106	1
HS48	2	5	0.00000e+00	0.00000e+00	0.00000e+00	3	0	0.005807	1
HS49	2	5	3.62439e-10	8.30436e-08	4.44089e-16	18	3	0.026359	1
HS50	3	5	1.51929e-64	1.28966e-17	6.28037e-16	10	0	0.009258	1
HS51	3	5	2.33974e-19	3.84452e-16	2.91421e-10	5	0	0.067605	1
HS52	3	5	5.32665e+00	8.88178e-16	1.47541e-11	6	0	0.006888	1
HS53	3	5	4.09302e+00	8.88178e-16	6.75581e-08	5	0	0.060449	1
HS56	4	7	-3.45600e+00	1.65648e-08	3.61299e-14	16	20	0.488529	1
HS6	1	2	5.04457e-17	7.09842e-09	5.90372e-11	4	2	0.079973	1
HS60	1	3	5.04457e-17	3.25834e-12	3.25834e-12	9	2	0.088096	1
HS61	2	3	-1.43646e+02	1.13687e-13	1.15879e-12	7	10	0.148419	1
HS7	1	2	-1.73205e+00	1.33338e-13	4.35207e-13	7	1	0.084446	1
HS77	2	5	2.41505e-01	2.22008e-08	1.17332e-09	10	2	0.110051	1
HS78	3	5	-2.91970e+00	3.49185e-08	3.71582e-12	7	1	0.084409	1
HS79	3	5	7.87768e-02	1.67192e-09	6.57084e-09	5	0	0.073266	1
HS8	2	2	-1.00000e+00	7.26249e-11	3.70990e-11	6	10	0.110542	1
HS9	1	2	-5.00000e-01	2.76909e-08	8.87042e-10	5	2	0.083123	1
HYDCAR20	99	99	0.00000e+00	7.13955e-12	2.54085e-13	23	120	1.304093	1
HYDCAR6	29	29	0.00000e+00	7.57366e-09	8.86869e-14	15	80	0.489626	1
HYPCIR	2	2	0.00000e+00	7.07905e-12	4.64338e-11	6	10	0.094463	1
INTEGREQ	500	502	0.00000e+00	1.20783e-08	1.22638e-07	4	0	2.251935	1
MARATOS	1	2	-1.00000e+00	4.19978e-11	8.37632e-11	6	0	0.066108	1
METHANB8	31	31	0.00000e+00	2.58376e-09	2.99989e-11	8	0	0.031527	1
METHANL8	31	31	0.00000e+00	1.45873e-08	3.87512e-13	11	22	0.403805	1
MSS1	73	90	-9.00000e+00	7.62993e-08	5.46212e-16	42	70	1.250480	1
MSS2	703	756	-4.80082e+02	2.76238e+00	2.94624e-01	6	1	10800	3
MWRIGHT	3	5	2.49788e+01	1.50090e-08	9.15513e-16	13	2	0.115340	1
ORTHREGB	6	27	2.20269e-26	3.37002e-12	1.46783e-14	6	0	0.078225	1
POWELLBS	2	2	0.00000e+00	1.20221e-16	2.27077e-16	19	120	0.370171	1
POWELLSQ	2	2	0.00000e+00	7.27007e-09	3.17057e-09	6	6	0.088129	1
RECIPE	3	3	0.00000e+00	3.53080e-09	9.32348e-08	15	0	0.087483	1
RSNBRNE	2	2	0.00000e+00	3.23758e-10	1.59191e-11	5	1	0.083472	1
S316-322	1	2	3.34315e+02	4.97131e-11	3.86133e-11	8	3	0.100933	1
SINVALNE	2	2	0.00000e+00	2.62161e-09	2.25440e-12	5	10	0.152583	1
ZANGWIL3	3	3	0.00000e+00	1.45923e-11	5.94197e-11	5	0	0.061785	1

6.2 Numerical Results of Primal-Dual Path-Following Shifted Penalty-Barrier Method

This section provides the numerical results of the primal-dual path-following shifted penalty-barrier method (PDPB) for (NIP) described in Chapter 5. Similar to the primal-dual path-following augmented Lagrangian method, this method has both the "outer" and "inner" iteration, defined by "OItr" and "FItr" in the table. The outer iterations update the multipliers estimate and adjust the penalty and barrier parameters depending on the feasibility and complementarity violations. In the inner iteration, a primal-dual shifted penalty-barrier merit function $\mathcal{M}(x, s, y, w; y^E, w^E, \mu^P, \mu^B)$ defined in Equation (5.15) is minimized to restrict the iterates to lie within current neighborhood of the trajectory. Problems in all-inequality form, i.e., in the form of (NIP) from CUTEst are tested.

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

Table 6.4 lists the default parameters. The specific meaning of each parameter is described in Algorithm 5.1 and Algorithm 5.2.

Table 6.4: Default Parameters Used in PDPB

Parameter	value	Parameter	value
a_P	0.5	δ_0	0.5
a_B	0.75	ρ_1	0.5
θ_μ	0.2	α_0	1.0
γ_s	0.5	η_1	0.5
γ_c	0.5	η_s	0.05
γ_e	2	η_e	0.25
y^E_{Max}	1.0e+6	tiny	1.0e-12
Tol	1.0e-5	MaxItn	1500

Both the conventional trust-region method and a combined trust-region line-search method have been applied for minimizing $\mathcal{M}(x, s, y, w; y^E, w^E, \mu^P, \mu^B)$ and their numerical performance are compared. A detailed description of these two methods can be referred to Algorithm 5.2 and Algorithm 5.3.

When minimizing the shifted barrier function where the logarithm term creates a "barrier" near boundary, the next iterate not only has to guarantee the sufficient decrease condition be satisfied but also has to be restricted in the interior of feasible region. Sometimes, the trust-region trial step is rejected simply because it falls into the infeasible region. In this case, it may be too expensive to reduce trust-region radius and wait for the next iterate to become feasible. Thus, line search strategy along the trust-region step that pulls iterates back into the interior of feasible region will be cheaper than stalling and continuously reducing the trust-region radius.

If the tested problems are in all-equality form, i.e., the slack variable $s = 0$, then the numerical results are given in Table 6.5 where conventional trust-region method is applied, and in Table 6.6 where the combined trust-region line-search method is applied. Otherwise, if the set $\mathcal{F} = \{x : c(x) > 0\}$ is nonempty, the numerical results are given in Table 6.5 and Table 6.6 respectively. It can be observed from those tables that the combined trust-region line-search method will generally require fewer matrix factorizations and fewer iterations than conventional trust-region method.

When $s = 0$, there will be no barrier term in the proposed primal-dual path-following shifted penalty-barrier method, so only the penalty term is retained. In this case, the proposed algorithm only fails to solve MSS2 due to the maximum running time limit is reached. In the case where $\mathcal{F} = \{x : c(x) > 0\}$ is nonempty, this algorithm efficiently solves most of the problems except a series of similar problems HS88, HS89, HS90, HS91 and HS92, where it stops at a infeasible stationary point in these five special cases.

Table 6.5: Numerical Results of PDPB – With Trust-Region Method

Prob	m	n	Opt-f(x)	Opt-Stny	$\|\|c(x^*)\|\|$	OItr	FItr	CPU(s)	Conv
ARGTRIG	200	200	0.00000e+00	7.43711e-11	7.99169e-09	3	4	0.952822	1
BT1	1	2	-9.99780e-01	5.99023e-08	2.20984e-06	6	751	0.861791	1
BT2	1	3	3.25682e-02	2.33638e-06	8.88559e-07	10	27	0.280940	1
BT3	3	5	4.09302e+00	4.44089e-16	1.90853e-07	2	8	0.171912	1
BT4	2	3	-4.55105e+01	4.02987e-08	2.64722e-07	2	53	0.195089	1
BT5	2	3	9.61715e+02	1.67374e-07	1.82205e-06	3	63	0.270413	1
BT6	2	5	2.77045e-01	1.36992e-08	1.92293e-09	9	23	0.294572	1
BT7	3	5	3.06038e+02	9.39800e-06	7.08547e-06	4	119	0.347058	1
BT8	2	5	1.00000e+00	2.74600e-06	4.67786e-06	8	10	0.234397	1
BT9	2	4	-1.00000e+00	2.48338e-08	2.69270e-07	6	9	0.181054	1
BT10	2	2	-1.00000e+00	1.31836e-06	7.23266e-15	5	6	0.221727	1
BT11	3	5	8.24885e-01	3.98395e-06	3.82443e-06	4	10	0.233964	1
BT12	3	5	6.18812e+00	3.70274e-13	2.11061e-11	4	34	0.255343	1
BOOTH	2	2	0.00000e+00	8.20186e-16	1.33227e-15	1	4	0.074317	1
BROWNALE	200	200	0.00000e+00	1.59080e-10	7.39680e-09	2	7	0.634235	1
BYRDSPHR	2	3	-4.68330e+00	1.72237e-09	1.90422e-06	5	96	0.351170	1
CHANDHEQ	100	100	0.00000e+00	9.21653e-07	1.79118e-06	7	11	0.452917	1
CHANDHEU	500	500	0.00000e+00	2.72999e-07	1.02360e-06	7	12	16.214911	1
CLUSTER	2	2	0.00000e+00	1.41126e-07	3.37438e-07	7	9	0.115730	1
CUBENE	2	2	0.00000e+00	1.60289e-16	1.11022e-15	3	25	0.199684	1
EIGMAXB	101	101	-9.67461e-04	3.04837e-08	4.51759e-08	6	256	5.822212	1
EIGMAXC	202	202	-1.00000e+00	1.23224e-07	3.88125e-13	3	6	1.071794	1
EIGMINB	101	101	9.67436e-04	8.17435e-12	1.33950e-10	7	247	30.551729	1
EIGMINC	202	202	1.00000e+00	3.14108e-09	1.27017e-07	3	6	1.069688	1
GENHS28	8	10	9.27174e-01	2.99349e-15	5.02258e-07	1	4	0.179981	1
GOTTFR	2	2	0.00000e+00	9.49538e-06	4.09181e-06	3	22	0.278952	1
HATFLDF	3	3	0.00000e+00	9.58511e-10	8.87424e-09	7	143	0.473387	1
HATFLDG	25	25	0.00000e+00	4.39815e-07	1.23535e-06	3	10	0.234190	1
HEART6	6	6	0.00000e+00	1.12858e-07	1.08957e-06	3	1690	2.117593	1
HEART8	8	8	0.00000e+00	5.16782e-10	2.87181e-08	2	133	0.328297	1
HIMMELBA	2	2	0.00000e+00	1.59941e-15	0.00000e+00	1	4	0.175033	1
HIMMELBC	2	2	0.00000e+00	1.75555e-06	5.29443e-06	1	6	0.158203	1
HIMMELBE	3	3	0.00000e+00	9.15623e-16	1.00000e-06	1	5	0.159197	1
HS100LNP	2	7	6.80630e+02	3.15786e-11	5.23143e-11	4	14	0.233470	1
HS111LNP	3	10	-4.77612e+01	5.78432e-06	4.14268e-06	8	60	0.278535	1
HS26	1	3	1.27430e-09	1.35845e-06	4.92545e-06	10	86	0.265542	1
HS27	1	3	4.00000e-02	4.24216e-11	3.94214e-08	15	122	0.526288	1
HS28	1	3	3.85186e-31	1.41307e-15	2.22045e-16	1	4	0.160659	1

Table 6.5: Numerical Results of PDPB – With Trust-Region Method (continued)

Prob	m	n	Opt-f(x)	Opt-Stny	$\|\|c(x^*)\|\|$	OItr	FItr	CPU(s)	Conv
HS39	2	4	-1.00000e+00	2.48338e-08	2.69270e-07	6	9	0.200949	1
HS40	3	4	-2.50000e-01	2.51283e-09	6.08638e-08	3	33	0.227671	1
HS42	2	4	1.38579e+01	4.69410e-08	2.32192e-07	3	13	0.212463	1
HS46	2	5	5.96648e-10	4.77149e-07	6.04114e-06	11	23	0.170300	1
HS47	3	5	-2.67123e-02	5.54922e-08	2.66841e-06	8	42	0.187677	1
HS48	2	5	0.00000e+00	3.65426e-15	0.00000e+00	1	5	0.180120	1
HS49	2	5	2.05086e-08	6.85506e-06	1.33227e-15	13	16	0.293868	1
HS50	3	5	1.91548e-18	3.04341e-10	2.92808e-09	7	10	0.208468	1
HS51	3	5	4.29979e-13	1.05241e-14	4.08939e-07	1	4	0.164597	1
HS52	3	5	5.32664e+00	1.53837e-15	4.77340e-07	2	5	0.243172	1
HS53	3	5	4.09302e+00	9.93014e-16	1.90853e-07	2	5	0.280386	1
HS56	4	7	-3.45600e+00	1.07917e-06	8.18490e-08	3	98	0.314760	1
HS6	1	2	3.65538e-14	3.82414e-07	6.54531e-07	10	143	0.624099	1
HS60	1	3	3.25682e-02	8.57865e-10	1.95316e-10	4	50	0.309392	1
HS61	2	3	-1.43646e+02	2.97752e-08	1.33597e-07	2	58	0.230313	1
HS7	1	2	-1.73205e+00	4.85740e-11	2.41995e-10	6	45	0.332307	1
HS77	2	5	2.41505e-01	2.42202e-09	6.81452e-10	9	24	0.280364	1
HS78	3	5	-2.91970e+00	3.41112e-09	4.67863e-09	3	36	0.195700	1
HS79	3	5	7.87768e-02	9.36802e-06	3.31779e-06	3	5	0.207189	1
HS8	2	2	-1.00000e+00	8.56648e-10	1.67585e-08	2	7	0.151906	1
HS9	1	2	-5.00000e-01	2.44869e-14	1.77636e-15	4	5	0.233552	1
HYDCAR20	99	99	0.00000e+00	6.94830e-10	2.70142e-12	15	3265	72.639923	1
HYDCAR6	29	29	0.00000e+00	1.06478e-09	1.23769e-08	9	520	2.172299	1
HYPCIR	2	2	0.00000e+00	4.88113e-06	5.81552e-06	1	5	0.181564	1
INTEGREQ	500	502	0.00000e+00	2.44596e-10	3.25265e-08	2	5	6.066171	1
MARATOS	1	2	-1.00000e+00	5.16311e-08	9.33994e-07	4	40	0.211696	1
METHANB8	31	31	0.00000e+00	2.06141e-07	1.18209e-06	3	4	0.254157	1
METHANL8	31	31	0.00000e+00	3.72736e-11	9.58332e-09	7	74	0.743951	1
MSS1	73	90	-1.60000e+01	6.32760e-09	2.06310e-15	19	1138	14.708731	1
MSS2	703	756	-4.70743e+02	5.76711e+02	2.60104e-01	4	722	10800	3
MWRIGHT	3	5	2.49788e+01	1.78438e-06	4.93969e-06	3	13	0.259208	1
ORTHREGB	6	27	1.67903e-17	1.25817e-07	2.55146e-12	10	281	1.089761	1
POWELLBS	2	2	0.00000e+00	2.11131e-06	5.74141e-10	7	146	3.681982	1
POWELLSQ	2	2	0.00000e+00	4.15097e-09	7.73387e-10	9	3453	2.781519	1
RECIPE	3	3	0.00000e+00	1.97045e-06	4.34753e-10	8	13	0.211220	1
RSNBRNE	2	2	0.00000e+00	8.89737e-17	0.00000e+00	3	25	0.252081	1
S316-322	1	2	3.34314e+02	1.89984e-09	1.28399e-06	6	76	0.249859	1
SINVALNE	2	2	0.00000e+00	5.84730e-15	2.87811e-10	2	27	0.194386	1
ZANGWIL3	3	3	0.00000e+00	3.38529e-15	3.81378e-15	2	9	0.167490	1

Table 6.5: Numerical Results of PDPB – With Trust-Region Method (continued)

Prob	m	n	Opt-f(x)	Opt-Stny	Opt-Comp	OItr	FItr	CPU(s)	Conv
EXPFITA	22	5	1.13661e-03	4.99541e-14	2.60840e-11	29	53	0.766353	1
EXPFITB	102	5	5.01937e-03	2.24062e-13	2.09330e-10	24	113	7.320574	1
EXPFITC	502	5	2.33013e-02	5.51434e-11	9.81935e-07	26	151	313.492405	1
GIGOMEZ1	3	3	-3.00000e+00	7.22371e-10	1.74803e-09	7	44	0.366754	1
GIGOMEZ2	3	3	1.95222e+00	4.09806e-09	7.50766e-09	9	17	0.325394	1
GIGOMEZ3	3	3	2.00000e+00	1.65262e-10	1.56760e-09	8	11	0.277596	1
HS10	1	2	-1.00000e+00	1.59009e-09	1.63575e-09	8	13	0.215553	1
HS100	4	7	6.80630e+02	4.48219e-11	6.74230e-10	10	26	0.331773	1
HS100MOD	4	7	6.78680e+02	1.63875e-14	5.41995e-14	26	44	0.594596	1
HS11	1	2	-8.49849e+00	3.90313e-09	4.00553e-09	6	11	0.242098	1
HS113	8	10	2.43062e+01	4.17379e-13	2.87606e-10	15	33	0.429074	1
HS12	1	2	-3.00000e+01	9.44711e-11	7.36180e-11	7	23	0.301156	1
HS22	2	2	1.00001e+00	7.86918e-10	1.82354e-08	5	9	0.237382	1
HS268	5	5	3.27418e-11	7.07829e-12	2.90682e-07	24	62	0.563898	1
HS29	1	3	-2.26274e+01	1.77636e-15	3.67949e-13	6	35	0.298755	1
HS43	3	4	-4.39999e+01	5.36912e-09	2.63859e-08	8	32	0.311707	1
HS88	5	10	5.71642e-18	2.88136e-09	2.16212e-03	5	10	0.263530	2
HS89	1	3	5.40608e-01	5.54237e-09	4.49043e-05	7	23	0.288065	2
HS90	1	4	5.47649e-01	4.95812e-09	8.68826e-05	7	42	0.393203	2
HS91	1	5	5.70836e-01	1.54280e-10	3.89287e-05	7	49	0.470243	2
HS92	1	6	4.50754e-01	3.09376e-09	6.42206e-05	5	39	0.484255	2
MADSEN	6	3	6.16429e-01	2.35452e-10	4.39974e-09	11	39	0.378771	1
MINMAXRB	4	3	-3.20909e-09	2.64233e-14	3.77907e-09	7	72	0.433615	1
S268	5	5	3.27418e-11	7.07829e-12	2.90682e-07	24	62	0.530553	1
SPIRAL	2	3	-7.38762e-13	1.16258e-10	2.69270e-16	9	131	0.497645	1
VANDERM1	199	100	0.00000e+00	1.77153e-13	1.19263e-11	30	58	27.813040	1
VANDERM2	199	100	0.00000e+00	1.77153e-13	1.19263e-11	30	58	27.597196	1
VANDERM3	199	100	0.00000e+00	3.75000e-12	1.29209e-11	20	44	20.435622	1
WOMFLET	3	3	-8.42719e-15	4.21885e-15	1.31224e-15	8	52	0.371955	1

Table 6.6: Numerical Results of PDPB – With Combined Trust-Region Line-Search Method

Prob	m	n	Opt-f(x)	Opt-Stny	$\|\|c(x^*)\|\|$	OItr	FItr	CPU(s)	Conv
ARGTRIG	200	200	0.00000e+00	8.06606e-09	7.99169e-09	3	4	1.272725	1
BT1	1	2	-9.99128e-01	2.59542e-07	8.76799e-06	5	1138	2.315680	1
BT2	1	3	3.25682e-02	1.29598e-09	5.03859e-10	13	40	0.291255	1
BT3	3	5	4.09302e+00	4.44089e-16	1.90853e-07	2	8	0.187089	1
BT4	2	3	-4.55105e+01	1.20004e-09	5.59273e-08	2	29	0.218116	1
BT5	2	3	9.61715e+02	5.31689e-13	3.63165e-11	3	32	0.217986	1
BT6	2	5	2.77045e-01	9.06976e-06	4.70017e-07	8	12	0.211533	1
BT7	3	5	3.60382e+02	3.65579e-06	4.90734e-06	5	15	0.193667	1
BT8	2	5	1.00000e+00	2.74600e-06	4.67786e-06	8	10	0.211385	1
BT9	2	4	-1.00000e+00	2.48338e-08	2.69270e-07	6	9	0.181574	1
BT10	2	2	-1.00000e+00	3.44373e-07	1.31836e-06	5	6	0.189296	1
BT11	3	5	8.24885e-01	1.25108e-09	7.45289e-08	5	9	0.214555	1
BT12	3	5	6.18812e+00	3.65028e-07	2.56367e-06	5	39	0.266790	1
BOOTH	2	2	0.00000e+00	8.20186e-16	1.33227e-15	1	4	0.113742	1
BROWNALE	200	200	0.00000e+00	1.59080e-10	7.39680e-09	2	7	0.704079	1
BYRDSPHR	2	3	-4.68330e+00	6.80735e-12	6.21694e-11	6	76	0.428139	1
CHANDHEQ	100	100	0.00000e+00	9.21653e-07	1.79118e-06	7	11	0.752851	1
CHANDHEU	500	500	0.00000e+00	2.72999e-07	1.02360e-06	7	12	16.127269	1
CLUSTER	2	2	0.00000e+00	1.41126e-07	3.37438e-07	7	9	0.165137	1
CUBENE	2	2	0.00000e+00	1.55411e-13	9.19242e-07	5	11	0.224762	1
EIGMAXB	101	101	-2.41391e-02	2.87081e-06	4.92425e-07	7	113	2.900165	1
EIGMAXC	202	202	-1.00000e+00	2.04706e-09	1.23224e-07	3	6	1.036842	1
EIGMINB	101	101	8.70136e-03	3.17435e-06	3.05175e-07	6	125	3.528658	1
EIGMINC	202	202	1.00000e+00	3.14108e-09	1.27017e-07	3	6	1.209293	1
GENHS28	8	10	9.27174e-01	2.99349e-15	5.02258e-07	1	4	0.197985	1
GOTTFR	2	2	0.00000e+00	7.71694e-06	3.75938e-06	3	7	0.238518	1
HATFLDF	3	3	0.00000e+00	5.05935e-09	2.46610e-10	7	66	0.365711	1
HATFLDG	25	25	0.00000e+00	1.43030e-08	1.36108e-07	4	8	0.181484	1
HEART6	6	6	0.00000e+00	5.55477e-12	4.52767e-11	3	677	1.504923	1
HEART8	8	8	0.00000e+00	3.34916e-07	7.06718e-07	2	39	0.211308	1
HIMMELBA	2	2	0.00000e+00	1.59941e-15	0.00000e+00	1	4	0.140900	1
HIMMELBC	2	2	0.00000e+00	1.75555e-06	5.29443e-06	1	6	0.149396	1
HIMMELBE	3	3	0.00000e+00	9.15623e-16	1.00000e-06	1	5	0.156458	1
HS100LNP	2	7	6.80630e+02	5.41698e-09	1.85731e-09	4	22	0.218618	1
HS111LNP	3	10	-4.77612e+01	2.21969e-07	1.52999e-07	9	54	0.283487	1
HS26	1	3	3.20965e-14	2.02687e-07	2.49163e-06	10	49	0.323904	1
HS27	1	3	4.00000e-02	1.42986e-09	4.91685e-07	10	70	0.382045	1
HS28	1	3	3.85186e-31	1.41307e-15	2.22045e-16	1	4	0.183492	1

Table 6.6: Numerical Results of PDPB – With Combined Trust-Region Line-Search Method (continued)

Prob	m	n	Opt-f(x)	Opt-Stny	$\|\|c(x^*)\|\|$	OItr	FItr	CPU(s)	Conv
HS39	2	4	-1.00000e+00	2.48338e-08	2.69270e-07	6	9	0.186183	1
HS40	3	4	-2.50000e-01	2.99075e-08	5.09329e-08	4	18	0.189881	1
HS42	2	4	1.38579e+01	7.54255e-06	2.28365e-06	3	9	0.175114	1
HS46	2	5	7.50615e-10	5.66017e-07	6.76763e-06	12	16	0.240622	1
HS47	3	5	5.71366e-10	2.88479e-06	7.22961e-07	7	21	0.288569	1
HS48	2	5	4.93038e-32	3.65426e-15	0.00000e+00	1	5	0.165841	1
HS49	2	5	2.05086e-08	6.85506e-06	1.33227e-15	13	16	0.229758	1
HS50	3	5	1.91548e-18	3.04341e-10	2.92808e-09	7	10	0.196628	1
HS51	3	5	4.29979e-13	1.05241e-14	4.08939e-07	1	4	0.140374	1
HS52	3	5	5.32664e+00	1.53837e-15	4.77340e-07	2	5	0.144989	1
HS53	3	5	4.09302e+00	9.93014e-16	1.90853e-07	2	5	0.184323	1
HS56	4	7	-3.45600e+00	6.50545e-08	9.16030e-09	3	57	0.269088	1
HS6	1	2	1.12189e-13	6.69930e-07	1.16893e-06	11	188	0.704800	1
HS60	1	3	3.25682e-02	2.40142e-06	9.34017e-07	4	39	0.276654	1
HS61	2	3	-1.43646e+02	1.97692e-07	2.57440e-07	2	40	0.190021	1
HS7	1	2	-1.73205e+00	2.32806e-10	9.09067e-10	6	28	0.276134	1
HS77	2	5	2.41505e-01	1.20612e-06	1.07632e-07	8	13	0.251527	1
HS78	3	5	-2.91970e+00	1.37457e-06	1.55949e-06	2	19	0.203791	1
HS79	3	5	7.87768e-02	9.36802e-06	3.31779e-06	3	5	0.174588	1
HS8	2	2	-1.00000e+00	8.56648e-10	1.67585e-08	2	7	0.162667	1
HS9	1	2	-5.00000e-01	2.44869e-14	1.77636e-15	4	5	0.204034	1
HYDCAR20	99	99	0.00000e+00	8.33355e-07	3.22427e-08	14	1839	46.196419	1
HYDCAR6	29	29	0.00000e+00	1.86043e-06	3.74133e-06	8	194	1.413919	1
HYPCIR	2	2	0.00000e+00	4.88113e-06	5.81552e-06	1	5	0.181564	1
INTEGREQ	500	502	0.00000e+00	2.44596e-10	3.25265e-08	2	5	5.037539	1
MARATOS	1	2	-1.00000e+00	7.69744e-10	4.25298e-09	4	23	0.205090	1
METHANB8	31	31	0.00000e+00	2.06141e-07	1.18209e-06	3	4	0.199214	1
METHANL8	31	31	0.00000e+00	1.81779e-09	1.30471e-08	5	11	0.280644	1
MSS1	73	90	-1.60000e+01	2.84308e-08	4.02030e-09	15	750	11.036012	1
MSS2	703	756	-4.70743e+02	5.76711e+02	2.60104e-01	4	722	10800	3
MWRIGHT	3	5	2.49788e+01	7.62659e-06	2.25925e-06	3	8	0.217571	1
ORTHREGB	6	27	1.96510e-14	9.90443e-06	8.83928e-09	8	260	1.173991	1
POWELLBS	2	2	0.00000e+00	1.11824e-08	2.87932e-11	14	156	6.415692	1
POWELLSQ	2	2	0.00000e+00	8.23021e-06	2.09012e-06	6	179	0.347545	1
RECIPE	3	3	0.00000e+00	1.97045e-06	4.34753e-10	8	13	0.218666	1
RSNBRNE	2	2	0.00000e+00	1.75346e-16	2.22045e-15	3	13	0.208964	1
S316-322	1	2	3.34314e+02	2.36584e-08	4.55391e-06	6	50	0.228420	1
SINVALNE	2	2	0.00000e+00	6.88650e-19	9.38963e-16	2	18	0.249397	1
ZANGWIL3	3	3	0.00000e+00	3.38529e-15	3.81378e-15	2	9	0.161238	1

Table 6.6: Numerical Results of PDPB – With Combined Trust-Region Line-Search Method(continued)

Prob	m	n	Opt-f(x)	Opt-Stny	Opt-Comp	OItr	FItr	CPU(s)	Conv
EXPFITA	22	5	1.13661e-03	4.99541e-14	2.60840e-11	29	53	0.802748	1
EXPFITB	102	5	5.01937e-03	2.24062e-13	2.09330e-10	24	113	5.884853	1
EXPFITC	502	5	2.33013e-02	5.51434e-11	9.81935e-07	26	151	310.223433	1
GIGOMEZ1	3	3	-3.00000e+00	3.34093e-10	1.23977e-09	6	50	0.399629	1
GIGOMEZ2	3	3	1.95222e+00	4.09806e-09	7.50766e-09	9	17	0.406706	1
GIGOMEZ3	3	3	2.00000e+00	1.65262e-10	1.56760e-09	8	11	0.320078	1
HS10	1	2	-1.00000e+00	1.59009e-09	1.63575e-09	8	13	0.266225	1
HS100	4	7	6.80630e+02	4.48219e-11	6.74230e-10	10	26	0.356232	1
HS100MOD	4	7	6.78680e+02	4.27177e-14	5.64757e-14	25	40	0.557153	1
HS11	1	2	-8.49849e+00	3.90313e-09	4.00553e-09	6	11	0.273735	1
HS113	8	10	2.43062e+01	4.17379e-13	2.87606e-10	15	35	0.404290	1
HS12	1	2	-3.00000e+01	3.79642e-13	5.00945e-11	6	19	0.277470	1
HS22	2	2	1.00001e+00	7.86918e-10	1.82354e-08	5	9	0.237382	1
HS268	5	5	3.27418e-11	7.07829e-12	2.90682e-07	24	62	0.563898	1
HS29	1	3	-2.26274e+01	1.32990e-13	1.51649e-12	5	22	0.319252	1
HS43	3	4	-4.39999e+01	6.70764e-09	5.89163e-08	8	26	0.360654	1
HS88	5	10	5.71642e-18	2.88136e-09	2.16212e-03	5	10	0.312474	2
HS89	1	3	5.40098e-01	5.79596e-09	4.50112e-05	7	18	0.391706	2
HS90	1	4	5.47648e-01	2.96035e-09	8.68831e-05	7	19	0.467704	2
HS91	1	5	4.72821e-01	2.10720e-11	5.83515e-05	5	20	0.445347	2
HS92	1	6	4.90790e-01	5.86898e-13	2.69504e-05	6	24	0.514359	2
MADSEN	6	3	6.16430e-01	1.97249e-10	4.30007e-09	11	41	0.534958	1
MINMAXRB	4	3	-4.52210e-16	1.40756e-10	2.27653e-08	6	46	0.410297	1
S268	5	5	3.27418e-11	7.07829e-12	2.90682e-07	24	62	0.632436	1
SPIRAL	2	3	-1.36224e-11	9.41038e-10	1.36102e-14	8	73	0.419347	1
VANDERM1	199	100	0.00000e+00	7.83180e-14	3.28268e-12	18	41	19.142302	1
VANDERM2	199	100	0.00000e+00	1.77153e-13	1.19263e-11	18	41	18.721808	1
VANDERM3	199	100	0.00000e+00	2.88437e-13	1.26707e-12	33	47	24.838323	1
WOMFLET	3	3	-2.00215e-10	3.06189e-12	1.87451e-12	6	34	0.423700	1

Bibliography

[1] R. Andreani, J. M. Martinez, A. Ramos, and P. J. Silva. A cone-continuity constraint qualification and algorithmic consequences. Optimization Online, 2015.

[2] P. Armand, J. Benoist, R. Omheni, and V. Pateloup. Study of a primal-dual algorithm for equality constrained minimization. *Computational Optimization and Applications*, 59(3):405–433, 2014.

[3] P. Armand and R. Omheni. A globally and quadratically convergent primal–dual augmented Lagrangian algorithm for equality constrained optimization. *Optimization Methods and Software*, 32(1):1–21, 2017.

[4] D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Automatic Control*, AC-21(2):174–184, 1976.

[5] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods.* Athena Scientific, Belmont, Massachusetts, 1996.

[6] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson. An estimate for the condition number of a matrix. *SIAM J. Numer. Anal.*, 16(2):368–375, 1979.

[7] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc.*, 49(1):1–23, 01 1943.

[8] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* John Wiley and Sons, Inc., New York-London-Sydney, 1968.

[9] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming.* Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 1990. Reprint of the 1968 original.

[10] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. Numerical Analysis Report 96-3, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1996.

[11] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. Optim.*, 8:1132–1152, 1998.

[12] A. Forsgren, P. E. Gill, and E. Wong. Primal and dual active-set methods for convex quadratic programming. *Math. Program.*, 159:460–508, 2016.

[13] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Rev.*, 44:525–597, 2002.

[14] A. L. Forsgren, P. E. Gill, and W. Murray. A modified Newton method for unconstrained minimization. Report SOL 89-12, Department of Operations Research, Stanford University, Stanford, CA, 1989.

[15] E. M. Gertz. *Combination Trust-Region Line-Search Methods for Unconstrained Optimization.* PhD thesis, Department of Mathematics, University of California, San Diego, 1999.

[16] E. M. Gertz. A quasi-Newton trust-region method. *Math. Program.*, 100(3, Ser. A):447–470, 2004.

[17] E. M. Gertz and P. E. Gill. A primal-dual trust-region algorithm for nonlinear programming. Numerical Analysis Report NA 02-1, University of California, San Diego, 2002.

[18] P. E. Gill, V. Kungurtsev, and D. P. Robinson. A stabilized sqp method: superlinear convergence. *Mathematical Programming*, 163(1):369–410, 2017.

[19] P. E. Gill, V. Kungurtsev, and D. P. Robinson. A shifted primal-dual penalty-barrier method for nonlinear optimization. 2018.

[20] P. E. Gill, V. Kungurtsev, and D. P. Robinson. A shifted primal-dual penalty-barrier method for nonlinear optimization. Center for Computational Mathematics Report CCoM 19-03, University of California, San Diego, 2019.

[21] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Primal-dual methods for linear programming. Report SOL 91-3, Department of Operations Research, Stanford University, Stanford, CA, 1991.

[22] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Solving reduced KKT systems in barrier methods for linear and quadratic programming. Report SOL 91-7, Department of Operations Research, Stanford University, Stanford, CA, 1991.

[23] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM J. Matrix Anal. Appl.*, 13:292–311, 1992.

[24] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47:99–131, 2005.

[25] P. E. Gill and D. P. Robinson. A primal-dual augmented Lagrangian. Numerical Analysis Report 08-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2008.

[26] P. E. Gill and E. Wong. Sequential quadratic programming methods. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 147–224. Springer New York, 2012.

[27] P. E. Gill and E. Wong. Methods for convex and general quadratic programming. *Math. Prog. Comp.*, 7:71–112, 2015.

[28] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Software*, 29(4):373–394, 2003.

[29] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads. Technical report, Rutherford Appleton Laboratory, Chilton, England, 2013.

[30] M. R. Hestenes. Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4:303–320, 1969.

[31] J. J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Math. Program.*, 16:1–20, 1979.

[32] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. and Statist. Comput.*, 4:553–572, 1983.

[33] S. G. Nash, R. Polyak, and A. Sofer. Numerical comparison of barrier and modified-barrier methods for large-scale bound-constrained optimization. In D. W. Hearn and P. M. Pardalos, editors, *Large-Scale Optimization: State of the Art*, pages 319–338. Kluwer, Dordrecht, 1994.

[34] J. Nocedal and Y.-X. Yuan. Combining trust region and line search techniques. In *Advances in Nonlinear Programming (Beijing, 1996)*, volume 14 of *Appl. Optim.*, pages 153–175. Kluwer Acad. Publ., Dordrecht, 1998.

[35] R. A. Polyak. Modified barrier functions (theory and methods). *Math. Program.*, 54(2, Ser. A):177–222, 1992.

[36] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298, London and New York, 1969. Academic Press.

[37] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114. Gordon and Breach, 1970.

[38] M. J. D. Powell. A new algorithm for unconstrained optimization. In J. B. Rosen, O. L. Mangasarian, and K. Ritter, editors, *Nonlinear Programming (Proc. Sympos., Univ. of Wisconsin, Madison, Wis., 1970)*, pages 31–65. Academic Press, New York, 1970.

[39] M. J. D. Powell. Convergence properties of a class of minimization algorithms. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming, 2 (Proc. Sympos. Special Interest Group on Math. Programming, Univ. Wisconsin, Madison, Wis., 1974)*, pages 1–27. Academic Press, New York, 1974.

[40] M. J. D. Powell. On the global convergence of trust region algorithms for unconstrained minimization. *Math. Programming*, 29(3):297–303, 1984.

[41] D. P. Robinson. *Primal-Dual Methods for Nonlinear Optimization.* PhD thesis, Department of Mathematics, University of California San Diego, La Jolla, CA, 2007.

[42] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, New Jersey, 1970.

[43] D. C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19(2):409–426, 1982.

[44] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57–88, London and New York, 1981. Academic Press.

[45] M. H. Wright. Ill-conditioning and computational error in interior methods for nonlinear programming. *SIAM J. Optim.*, 9(1):84–111, 1998.