UNIVERSITY OF CALIFORNIA SAN DIEGO

Primal-Dual Trust-Region Methods For Nonlinear Programming

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

in

Mathematics

by

Yesheng Huang

Committee in charge:

Professor Philip E. Gill, Chair
Professor Randolph Edwin Bank
Professor Michael Holst
Professor Ronghui Xu
Professor Danna Zhang

2023

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

University of California San Diego

2023

DEDICATION

My best friend in the past year, Shovel.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This thesis is a witness of the help and supports I have received over the years, during which I had been combating serious mental problems, which made me slow in making progress.

Firstly, I have to pay gratitude to my advisor, Professor Philip Gill for his instructions as well as his patience and tolerance with me. He helped me through many academic and administrative affairs during possibly the hardest time in my life.

Secondly, I am very grateful to Mr. Scott Rollans and Mr. Mark Whelan, for their help and support in administrative affairs. They, along with Prof.Gill, helped me figure out how to apply for extension due to my illness.

Thirdly, I am thankful to the committee for their time to review and assess.

Finally, I want to thank Shovel, a maltipoo dog(the cross breed between maltese and poodle), who is actually not even mine, for the time we spent together and the comfort he offered in the past year.

VITA

2015 Bachelor of Science in Mathematics, Fudan University

2023 Doctor of Philosophy in Mathematics, University of California San Diego

ABSTRACT OF THE DISSERTATION

Primal-Dual Trust-Region Methods For Nonlinear Programming

by

Yesheng Huang

Doctor of Philosophy in Mathematics

University of California San Diego, 2023

Professor Philip E. Gill, Chair

The goal of this dissertation is to investigate the formulation and analysis of a trust-region interior-point method for solving nonconvex optimization problems with a mixture of equality and inequality constraints. The proposed method is based on minimizing a merit function that may be interpreted as a shifted primal-dual penalty-barrier function. The method generates a sequence of iterates with limit points that are either infeasible stationary points or complementary approximate Karush-Kuhn-Tucker points, i.e., every limit point satisfies reasonable stopping criteria and is a Karush-Kuhn-Tucker point under a regularity condition that is the weakest constraint qualification associated with sequential optimality conditions. Under suitable additional assumptions, the method is equivalent to a shifted variant of the primal-dual

path-following method in the neighborhood of a solution.

The proposed method has an inner/outer iteration structure. The outer iteration specifies the form of the merit function. The inner iteration optimizes the merit function with fixed parameters using a trust-region method. The algorithm for solving the trust-region subproblem involves a procedure based on the application of a one-dimensional Newton's method. Methods are proposed for treating the so-called "hard case" in which no root of the one dimensional equation exists.

Chapter 1

Introduction

1.1 Overview

An optimization problem begins with a set of independent variables, and often includes conditions or restrictions that define acceptable values of the variables. Such restrictions are known as the constraints of the problem. The other essential component of an optimization problem is a single measure of "goodness", termed the objective function, which depends in some way on the variables. The solution of an optimization problem is a set of allowed values of the variables for which the objective function assumes its "optimal" value. In mathematical terms, this usually involves maximizing or minimizing. The optimization problem considered in this thesis has the form:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i \in \mathscr{E}, \\ & c_i(x) \geq 0, \quad i \in \mathscr{I}, \end{aligned}$$

where it is assumed that the scalar-valued functions f and $\{c_i(x)\}$ are at least twice-continuously differentiable.

One special case is that there are no constraints at all. In this case, the problem is called unconstrained. Two of the most widely used methods to solve unconstrained problems are line-search methods and trust-region methods. Both are iterative and model-based. Line-search methods first compute a descent direction, and then find a suitable step along this direction that

gives a sufficient decrease of the objective function. Trust-region methods find a minimizer of a model of the objective function subject to a constraint that attempts to define a neighborhood in which the model may be "trusted" as a good approximation of the objective function. The size of this neighborhood is reduced or enlarged depending on whether the approximate minimizer gives a sufficient decrease of the objective function.

Unconstrained problems are important in the sense that constrained problems can be transformed into a sequence of unconstrained problems. Constrained optimization problems can be classified into nonlinear equality constrained problems (NEP) and nonlinear inequality constrained problems (NIP). One popular approach is to incorporate constraints into the objective function. For equality constrained problems and inequality constrained problems, penalty terms and barrier terms can be added respectively to the objective functions. Both of these approaches associate a high cost for violating the constraints.

1.2 Contributions of This Dissertation

Chapter 2 concerns background material in optimization and introduces methods that generalize the penalty-function method of Gill & Robinson [14], and the penalty-barrier method of Gill, Kungurtsev & Robinson [13]. These methods inherit some features of augmented Lagrangian methods and modified barrier methods. In particular, it is not necessary for the penalty parameter to go to infinity or the barrier parameter to go zero. The algorithm treats the primal and dual variables as being independent and updates them simultaneously. At each iteration, a model-based subproblem is solved using a trust-region method.

Chapter 3 investigates numerical methods to solve a trust-region subproblem in which the trust-region constraint involves an elliptic norm. Particular attention is paid to degenerate cases and safeguarded methods are adopted to guarantee stability.

Chapter 4 describes a shifted primal-dual penalty-barrier function with slacks on constraints introduced in Gill, Kungurtsev & Robinson [13]. In this method, line-search method is

used to minimize a sequence of unconstrained optimization problems.

In Chapter 5 a primal-dual penalty-barrier function is proposed that uses shifts on the constraints instead of slacks.

1.2.1 Notation and terminology

Given vectors x and y, the vector consisting of x augmented by y is denoted by (x,y). The subscript i is appended to vectors to denote the i-th component of that vector, whereas the subscript k is appended to a vector to denote its value during the k-th iteration of an algorithm, e.g., x_k represents the value for x during the k-th iteration, whereas compx_{ki} denotes the i-th component of the vector x_k. Given vectors a and b with the same dimension, the vector with i-th component $a_i b_i$ is denoted by $a \cdot b$. Similarly, $\min(a,b)$ is a vector with components $\min(a_i,b_i)$. The vector e denotes the column vector of ones, and I denotes the identity matrix. The dimensions of e and I are defined by the context. The vector two-norm or its induced matrix norm are denoted by $\|\cdot\|$. The inertia of a real symmetric matrix A, denoted by $\text{In}(A)$, is the integer triple (a_+,a_-,a_0) giving the number of positive, negative and zero eigenvalues of A. The vector $g(x)$ is used to denote $\nabla f(x)$, the gradient of $f(x)$. The matrix $J(x)$ denotes the $m \times n$ constraint Jacobian, which has i-th row $\nabla c_i(x)^{\mathrm{T}}$. Let $\{\alpha_j\}_{j\geq 0}$ be a sequence of scalars, vectors, or matrices and let $\{\beta_j\}_{j\geq 0}$ be a sequence of positive scalars. If there exists a positive constant γ such that $\|\alpha_j\| \leq \gamma\beta_j$, we write $\alpha_j = O(\beta_j)$. If there exists a sequence $\{\gamma_j\} \to 0$ such that $\|\alpha_j\| \leq \gamma_j\beta_j$, we say that $\alpha_j = o(\beta_j)$. If there exists a positive sequence $\{\sigma_j\} \to 0$ and a positive constant β such that $\beta_j > \beta\sigma_j$, we write $\beta_j = \Omega(\sigma_j)$.

Chapter 2

Background

2.1 Unconstrained Optimization

This section gives a brief review of methods for unconstrained optimization. Methods for unconstrained problems are basis for many constrained optimization methods.

2.1.1 Preliminaries

Definition of a minimizer

In an unconstrained problem, a function $f : \mathscr{D} \subset \mathbb{R}^n \to \mathbb{R}$ is to be minimized within in the domain $\mathscr{D}$ of f.

Definition 2.1.1. *Given a continuous function $f : \mathscr{D} \subset \mathbb{R}^n \to \mathbb{R}$, $x^* \in \mathscr{D}$ is a global unconstrained minimizer of f on $\mathscr{D}$ if $f(x^*) \leq f(x)$ for all $x \in \mathscr{D}$. If x^* is a global unconstrained minimizer, then $f(x^*)$ is the global unconstrained minimum.*

Unless f is convex, the problem of finding a global minimizer is generally intractable. Instead, most methods focus on finding a point that minimizes f on some neighborhood of that point.

Definition 2.1.2. *Given $f : \mathscr{D} \to \mathbb{R}$, x^* is a local unconstrained minimizer of f if there exists an open ball $\mathscr{B}(x^*, \delta)$, such that $f(x^*) \leq f(x)$ for all $x \in \mathscr{B}(x^*, \delta)$.*

In what follows, all minimizers are meant to be local minimizers.

Optimality Conditions

In general, the verification of optimality involves comparing $f(x^*)$ with every value of f in the neighborhood of x^* is impossible. However, when f is sufficiently smooth, there are ways to determine whether x^* is a minimizer or not, using its derivatives.

The first result gives a necessary condition for a local minimizer when f is differentiable at x^*.

Theorem 2.1.1. *Given $f : \mathscr{D} \to \mathbb{R}$, assume that x^* is a minimizer and f is differentiable at x^*. Then $\nabla f(x^*) = 0$.*

Below a necessary condition and a sufficient condition are given when f has second derivatives at x^*.

Theorem 2.1.2. *Given $f : \mathscr{D} \to \mathbb{R}$, assume that x^* is an unconstrained minimizer of f, and f has second derivatives at x^*. Then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.*

Theorem 2.1.3. *Given $f : \mathscr{D} \to \mathbb{R}$, and f has second derivatives at x^*. Assume $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, then x^* is an unconstrained minimizer.*

2.1.2 Two Basic Optimization Methods

In most cases, a search direction is found by minimizing a local model. This direction is used to determine a new point that gives a sufficient decrease in the objective function. Line-search and trust-region methods are two of the most widely used approaches. Minimizers of the local model must exist and be easy to compute, and solutions of the local models should heuristically approach the solution of the original problem. For simplicity in this section, the local model is assumed to be

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

where H_k is the symmetric positive definite approximation of $\nabla^2 f(x_k)$, and the difference of $q_k(x_k)$ and $q_k(x_k+d)$ is denoted by

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

Line-Search Methods

In a line-search method, a search direction is computed (e.g., the steepest-descent direction) and a scalar step along this direction is adjusted to ensure that there is a sufficient decrease in the objective function after taking the step. The scalar γ_c in the algorithm is a contraction factor that is used to decrease the step length when there is not sufficient decrease in the objective function, and η_s is the reduction factor in that the step length will be reduced if the actual decrease in objective function is less than η_s times the reduction in the model function. Let p_k be the descent direction found at the k-th iteration. After each inner iteration, the inequality

$$f(x_k) - f(x_k + \alpha_k p_k) \geq \eta_s \big(q_k(x_k) - q_k(x_k + \alpha_k p_k) \big)$$

holds, which means sufficient decrease. The algorithm is given in Algorithm 2.1

Algorithm 2.1. Basic Line-Search Algorithm.

procedure LINE SEARCH(x_0)
  Specify $0 < \eta_s, \gamma_c < 1$;
  $k \leftarrow 0$;
  while not converged **do**
    $p_k = \arg\min\{q_k(x_k+d)\}$;
    $\alpha_k \leftarrow 1$;
    $\rho_k = \big(f(x_k) - f(x_k+\alpha_k p_k)\big) / \big(q(x_k) - q(x_k+\alpha_k p_k)\big)$;
    while $\rho_k < \eta_s$ **do**
      $\alpha_{k+1} \leftarrow \gamma_c \alpha_k$;
    end while
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
    $k \leftarrow k+1$;
  end while
end procedure

Trust-Region Methods

Trust-region methods explicitly limit the length of the step by defining d_k as an approximate solution of the constrained minimization problem:

$$\underset{d\in\mathbb{R}^n}{\text{minimize}}\ \ q_k(x_k+d) \quad \text{subject to} \quad \|d\| \le \delta_k.$$

Once the trust-region step d_k has been computed, the ratio of the actual and predicted reduction in f is computed as

$$\rho_k = \frac{f(x_k) - f(x_k+d_k)}{q_k(x_k) - q_k(x_k+d_k)}. \tag{2.1}$$

As long as the actual reduction is at least η_A times the predicted reduction, i.e., $\rho_k \ge \eta_A$, then x_k+d_k is selected as the new point x_{k+1}. If the test fails, i.e., if $\rho_k < \eta_A$, the trust-region radius is decreased by a contraction factor γ_c and the current trust-region subproblem is terminated with $x_{k+1} = x_k$. This strategy is based on the observation that the value of $q_k(x_k+d_k)$ for the next subproblem will be a better approximation to $f(x_k+d_k)$. It may be necessary to re-solve the subproblem several times before the predicted and actual reductions are comparable. It is important to note that, unlike a line-search method, the variables do not necessarily change at every iteration. The algorithm is given in Algorithm 2.2.

Algorithm 2.2. Basic Trust-Region Algorithm.

Specify constants $0 < \eta_A < \eta_E < 1$, $0 < \eta_A < \frac{1}{2}$, $0 < \gamma_c < 1 < \gamma_e$;
$k \leftarrow 0$; $\delta_k \leftarrow 1$;
while not *converged* **do**
  Compute d_k as an approximate solution of $\min_d \{q_k(x_k+d) : \|d\| \le \delta_k\}$;
  $\rho_k \leftarrow \big(f(x_k) - f(x_k+d_k)\big) / \big(q_k(x_k) - q_k(x_k+d_k)\big)$;
  if $\rho_k \ge \eta_A$ **then**
    $x_{k+1} \leftarrow x_k + d_k$; [Successful iteration.]
    if $\rho_k \ge \eta_E$ **then** $\delta_{k+1} = \max\ \{\ \delta_k, \gamma_e\|d_k\|\ \}$ **else** $\delta_{k+1} \leftarrow \delta_k$ **end if**
  else
    $x_{k+1} \leftarrow x_k$; $\delta_{k+1} \leftarrow \gamma_c\|d_k\|$;
  end if
  $k \leftarrow k+1$;
end while

For computational efficiency, the trust-region subproblem is not solved exactly. The amount of computation is limited without compromising the overall convergence. One concept here is called the Cauchy step. A step d_k^C is called Cauchy step if it's the solution to the trust-region subproblem

$$Q_k(d_k^C) = \min_{d,\alpha}\{Q_k(d)\colon d = -\alpha\nabla f(x_k), ||d|| \leq \delta_k\}. \tag{2.2}$$

It can be calculated that $d_k^C = -\alpha_k^* g_k$, where

$$\alpha_k^* = \begin{cases} g_k^{\mathrm{T}} g_k / g_k^{\mathrm{T}} H_k g_k & g_k^{\mathrm{T}} g_k / g_k^{\mathrm{T}} H_k g_k \leq \delta_k / ||g_k||, \text{ and } g_k^{\mathrm{T}} H_k g_k > 0, \\ \delta_k / ||g_k|| & \text{otherwise,} \end{cases} \tag{2.3}$$

where $g_k = -\nabla f(x_k)$. One property of the Cauchy step is given in the following lemma.

Lemma 2.1.4. *Given any norm $||\cdot||$, let κ be a constant such that $||d||_2 \geq \kappa||d||$ for all d. If d_k^C is the Cauchy step, then*

$$Q_k(d_k^C) \leq -\frac{1}{2}\kappa^2 ||g_k|| \min\{\delta_k, ||g_k||/||H_k||_2\}.$$

Convergence for trust-region methods depends on how accurate d_k is a s a solution to the trust-region algorithm. One weak requirement, proposed by Powell [29], requires for some $\tau > 0$ the following holds:

$$-Q_k(d_k) \geq \tau ||g_k|| \min\{\delta_k, ||g_k||/||H_k||\}, \text{ and } ||d_k|| \leq \delta_k. \tag{2.4}$$

Theorem 2.1.5 (Gill & Wright [17])**.** *Let $f\colon \mathscr{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on the open convex set $\mathscr{D}$. Let $\{x_k\} \subset \mathscr{D}$ be a sequence of iterates generated by the basic trust-region method of the above algorithm. Assume that an approximate solution d_k of the subproblem satisfies* (2.4)*. Assume further that the sequence $\{||H_k||\}$ is bounded. If f is bounded below in $\mathscr{D}$, then either* $\liminf_{k\to\infty} ||\nabla f(x_k)|| = 0$ *or some x_k satisfies the algorithm's convergence criterion and the algorithm terminates.*

2.1.3 Combination of Trust Region and Line Search

In Nocedal & Yuan [27], it was proposed to use a combination of trust region and line search method. Trust region has the advantage that it can handle ill-conditioned cases, but in each iteration, it's expensive to solve the trust region equation. The basic idea is that, in the trust region iteration, if the step does not give sufficient decrease of the function, the algorithm switches to line search method rather than change the trust region radius, which makes the computation faster. Gertz [11] proposed a backtracking method based on Armijo condition and Wolfe condition.

Gertz & Gill [12] proposed a combined trust region and line search method for use with a primal-dual interior method. The trust-region subproblem is

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & Q(x) \\ \text{subject to} \quad & ||x||_T \leq \delta. \end{aligned} \tag{2.5}$$

The Armijo-style condition is

$$f(v) - f(v+\alpha s) \geq -\eta_1 Q(\alpha s)), \tag{2.6}$$

where T is a positive definite matrix and $||s||_T := (s^{\mathrm{T}} T s)^{\frac{1}{2}}$ and δ is the trust-region radius.

2.2 Constrained Optimization

As said in the beginning chapter, the goal is to solve problems of the form:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i \in \mathscr{E}, \\ & c_i(x) \geq 0, \quad i \in \mathscr{I}, \end{aligned} \tag{2.7}$$

where it's assumed that both f and c_i's are at least twice continuously differentiable.

The first part goes through some commonly used definitions and results in nonlinear optimizations, along with the notations will be used. Later parts introduce methods used to

solve optimization problems that motivate the algorithm proposed in this thesis. Although our ultimate aim is to deal with constraints that involve both equalities and inequalities, it would be good to first investigate them separately, because they require different methods that contribute to the general problems. Methods that motivate the algorithm of this thesis will be discussed respectively in these parts.

2.2.1 Preliminaries

The gradient of f is defined to be a column vector as follows:

$$g(x) = \nabla f(x).$$

The Jacobian matrix related to the constraints is a matrix

$$J(x) = \begin{pmatrix} \nabla c_1(x)^{\mathrm{T}} \\ \nabla c_2(x)^{\mathrm{T}} \\ \vdots \\ \nabla c_i(x)^{\mathrm{T}} \\ \vdots \\ \nabla c_{(||\mathscr{E}||+||\mathscr{I}||)}(x)^{\mathrm{T}} \end{pmatrix},$$

where $i = 1, 2, \ldots, ||\mathscr{E}|| + ||\mathscr{I}||$.

The Hessian of the objective function is denoted by

$$H(x) = \nabla^2 f(x).$$

The Lagrangian function is defined as

$$L(x,y) = f(x) - \sum_{i \in \mathscr{E} \cup \mathscr{I}} y_i c_i(x),$$

and the Hessian of the Lagrangian function with respect to x is given by

$$H(x,y) = H(x) - \sum_{i \in \mathscr{E} \cup \mathscr{I}} y_i \nabla^2 c_i(x).$$

Below are definitions of minimizers for constrained optimization problems. These definitions are similar to those for unconstrained problems, but require an additional requirement that the optimal point satisfies the constraints.

Definition 2.2.1 (Feasible Region)**.** *The feasible region is defined to be the set*

$$\mathscr{F} := \{x \in \mathbb{R}^n \mid c_i(x) = 0, \quad \forall i \in \mathscr{E}, \qquad c_i(x) \geq 0, \quad \forall i \in \mathscr{I}\}.$$

Definition 2.2.2 (Feasible Path)**.** *A feasible path is a directed twice-differentiable curve $x(\alpha)$ starting at a feasible point x, such that*

- *$x(0) = x$ and $x(\alpha)$ is feasible for all $0 \leq \alpha < \hat{\alpha}$, for some $\hat{\alpha} > 0$.*
- *$\frac{d}{d\alpha}x(\alpha)\big|_{\alpha=0}$ is a nonzero vector.*

Definition 2.2.3 (Constrained Global Minimizer)**.** *A point x^* is called a constrained global minimizer, if $x^* \in \mathscr{F}$ and*

$$f(x^*) \leq f(x), \quad \forall x \in \mathscr{F}.$$

Furthermore, $f(x^*)$ the called a global minimum of f.

Definition 2.2.4 (Constrained Local Minimizer)**.** *A point x^* is called a constrained local minimizer if $x^* \in \mathscr{F}$ and there exists $\delta > 0$ such that*

$$f(x^*) \leq f(x), \quad \forall x \in \mathscr{F} \cap \mathscr{B}(x^*, \delta).$$

For constrained optimization, the constraints are expected to satisfy certain regularity assumptions called constraint qualifications, under which optimality conditions can be easily characterized. These assumptions are about cones, which are important geometric concepts in optimizations. Some related definitions are given below.

Definition 2.2.5 (Cone)**.** *A set $\mathscr{C}$ is called a cone if $\forall x \in \mathscr{C}, \forall \theta \geq 0 \Rightarrow \theta x \in \mathscr{C}$.*

Definition 2.2.6 (Convex Cone)**.** *A set $\mathscr{C}$ is a convex cone if it's a cone and convex.*

Definition 2.2.7 (Dual Cone)**.** *The dual cone $\mathscr{C}^*$ of $\mathscr{C}$ is defined as $\mathscr{C}^* = \{y \mid x^{\mathrm{T}} y \geq 0,\ \forall x \in \mathscr{C}\}$.*

It's obvious $\mathscr{C}^*$ is a convex cone.

Definition 2.2.8 (Tangent Cone)**.** *A tangent cone at a feasible point x is*

$$\mathscr{T}(x) = \left\{ p \mid \exists z_n \, feasible and t_n \searrow 0,\ p = \lim_{n\to\infty} \frac{z_n - x}{t_n} \right\}.$$

Definition 2.2.9 (Linear Cone)**.** *A linear cone at a feasible point x is*

$$\mathscr{T}_L(x) = \{p \mid \nabla c_i(x)^{\mathrm{T}} p = 0, \forall i \in \mathscr{E}, \quad \nabla c_i(x)^{\mathrm{T}} p \geq 0, \forall j \in \mathscr{I}_0(x)\}.$$

It's easy to see x^* is a local minimizer only if $\nabla f(x^*)^{\mathrm{T}} p \geq 0$ for all $p \in \mathscr{T}(x^*)$, which implies $\nabla f(x^*) \in \mathscr{T}^*(x^*)$. But this is hard to verify, so it's usually assumed that $\mathscr{T}_L^*(x^*) = \mathscr{T}^*(x^*)$ and the only thing need to check is whether or not $\nabla f(x^*) \in \mathscr{T}_L^*(x^*)$.

This leads to a constraint qualification (the weakest one that one can expect):

Definition 2.2.10 (Guignard Constraint Qualification (GCQ))**.** *GCQ holds at a feasible point x if $\mathscr{T}^*(x) = \mathscr{T}_L^*(x)$.*

Two of the most widely used constraint qualifications are LICQ and MFCQ, as given below.

Definition 2.2.11 (MFCQ)**.** *Let x^* be a feasible point of the program* (2.7)*. The Mangasarian-Fromovitz constraint qualification (MFCQ) holds at x^* if the gradient vectors*

$$\nabla c_i(x), \quad i \in \mathscr{E}$$

are linearly independent and there exists a vector $d \in \mathbb{R}^n$ such that

$$\nabla c_i(x)^{\mathrm{T}} d < 0, \quad \forall i \in \mathscr{I}_0,$$
$$\nabla c_i(x)^{\mathrm{T}} d = 0, \quad \forall i \in \mathscr{E}.$$

Definition 2.2.12 (LICQ)**.** *Let x^* be a feasible point of the program* (2.7)*. The linear independence constraint qualification (LICQ) holds at x^* if the constraint gradient vectors*

$$\nabla c_i(x^*), \quad i \in \mathscr{E} \cup \mathscr{I}_0$$

are linearly independent, where

$$\mathscr{I}_0 := \{i \in \mathscr{I} \mid c_i(x) = 0\}$$

is the active set of inequality constraints.

Proposition 2.2.1. *There is the implication*

$$LICQ \Rightarrow MFCQ \Rightarrow GCQ.$$

2.2.2 Problems with Equality Constraints

This part focuses on the treatment of *equality constraints*. In the most general case, the optimization problem to be considered is a nonlinear equality constrained problem (NEP), which is written in the form:

$$\underset{x \in \mathscr{D} \subseteq \mathbb{R}^n}{\text{minimize}} \; f(x) \quad \text{subject to} \; \; c_i(x) = 0, \quad i = 1, 2, \ldots, m, \tag{NEP}$$

where each $c_i(x)$ is a (possibly) nonlinear function of the n variables x_1, x_2, ..., x_n. For convenience, we will often consider the vector-valued function $c(x)$ that has the constraint function $c_i(x)$ as its i-th component, i.e., $c : \mathscr{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}^m$.

Now, optimality conditions can be given. More details can be found in Gill & Wright [17].

Definition 2.2.13 (KKT point for (NEP))**.** *A feasible point x^* such that $\nabla f(x^*) = J(x^*)^{\mathrm{T}} y^*$ for some m-vector y^* is called a* first-order KKT point *(or just a* KKT point*) for* (NEP)*. Equivalently, if the columns of $Z(x^*)$ form a basis for the null-space of $J(x^*)$, then x^* is a KKT point if $Z(x^*)^{\mathrm{T}} \nabla f(x^*) = 0$.*

Theorem 2.2.2 (First-order necessary optimality conditions for (NEP))**.** *If a constraint qualification holds at x^*, then x^* is a local solution of* (NEP) *only if x^* is a KKT point.*

Theorem 2.2.3 (Second-order necessary conditions for (NEP))**.** *If a constraint qualification holds at x^*, then x^* is a local solution of* (NEP) *only if:*

(a) *x^* is feasible, i.e., $c(x^*) = 0$;*

(b) *there exists a vector y^* such that $\nabla f(x^*) = J(x^*)^{\mathrm{T}} y^*$; and*

(c) *for the y^* of part* **(b)**, *$p^{\mathrm{T}} H(x^*, y^*) p \geq 0$ for every vector p satisfying $J(x^*)p = 0$.*

Theorem 2.2.4 (Sufficient conditions for a strict local minimizer)**.** *A point x^* is a strict local minimizer of* (NEP) *if*

(a) *x^* is feasible, i.e., $c(x^*) = 0$;*

(b) *there exists a vector y^* such that $\nabla f(x^*) = J(x^*)^{\mathrm{T}} y^*$; and*

(c) *for the y^* of part* **((b))**, *the strict inequality $p^{\mathrm{T}} H(x^*, y^*) p > 0$ holds for every $p \neq 0$ such that $J(x^*)p = 0$.*

2.2.3 The Method of Newton-Lagrange

When minimizing a function without constraints, a standard approach is to use the first-order optimality conditions to define a system of nonlinear equations $\nabla f(x) = 0$ whose solution is a first-order optimal point x^*. In the constrained case, the relevant system involves the gradient of the Lagrangian, which expresses the first-order feasibility and optimality conditions satisfied by x^* and y^*. Theorem 2.2.2 implies that the $n+m$ vector (x^*, y^*) is a solution of the $n+m$ nonlinear equations $F(x,y) = 0$, where

$$F(x,y) = \nabla L(x,y) = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ -c(x) \end{pmatrix}. \tag{2.8}$$

It follows that one approach to finding (x^*, y^*) is to apply Newton's method to find a zero of F. Let v denote an $(n+m)$-vector (x,y). Given an initial estimate $v_0 = (x_0, y_0)$ of a zero $v^* = (x^*, y^*)$, Newton's method generates a sequence of iterates v_k such that

$$v_{k+1} = v_k + \alpha_k \Delta v_k, \text{ where } F'(v_k)\Delta v_k = -F(v_k).$$

It only remains to compute the $(n+m) \times (n+m)$ Jacobian $F'(v)$. Differentiating (2.8) with respect to x and y gives $F'(v) = F'(x,y)$ as

$$F'(x,y) = \begin{pmatrix} \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 c_i(x) & -J(x)^{\mathrm{T}} \\ -J(x) & 0 \end{pmatrix}.$$

Using this Jacobian in the Newton equations gives

$$\begin{pmatrix} H(x_k, y_k) & -J(x_k)^{\mathrm{T}} \\ -J(x_k) & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) - J(x_k)^{\mathrm{T}} y_k \\ -c(x_k) \end{pmatrix}.$$

The algorithm of Newton-Lagrange is given in Algorithm 2.3.

Algorithm 2.3. Method of Newton-Lagrange.

Fix η_A and γ_c such that $0 < \eta_A < \frac{1}{2}$ and $0 < \gamma_c < 1$;
Choose x_0 and y_0;
$x \leftarrow x_0$; $y \leftarrow y_0$;
$k \leftarrow 0$;
while not *converged* **do**
  Solve the KKT system $\begin{pmatrix} H(x,y) & J(x)^{\mathrm{T}} \\ J(x) & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ c(x) \end{pmatrix}$;
  $\alpha \leftarrow 1$;
  while $\|F(v_k)\| - \|F(v_k + \alpha_k \Delta v_k)\| < \eta_A \big(\|M_k(v_k)\| - \|M_k(v_k + \alpha_k \Delta v_k)\| \big)$ **do**
    $\alpha \leftarrow \gamma_c \alpha$;
  end while
  $x \leftarrow x + \alpha \Delta x$; $y \leftarrow y + \alpha \Delta y$;
  $k \leftarrow k + 1$;
end while

Penalty Method

One method is to solve a sequence of unconstrained problems by adding constraints to the objectives. Those added terms are called penalties. Parameters associated with the penalties are to be increased dynamically to enforce feasibility. One popular penalty method is the quadratic penalty method, which is given as follows:

$$P_2(x;\rho) = f(x) + \tfrac{1}{2}\rho \sum_{i=1}^{m} c_i(x)^2 = f(x) + \tfrac{1}{2}\rho c(x)^{\mathrm{T}} c(x) = f(x) + \tfrac{1}{2}\rho \|c(x)\|_2^2, \tag{2.9}$$

where the nonnegative scalar ρ is called the *penalty parameter*. In the "classical" penalty-function method, $P_2(x;\rho)$ is minimized for each of a sequence of increasing values of ρ using an unconstrained minimization method. Let $x(\rho)$ denote an unconstrained minimizer of $P_2(x;\rho)$. The main property expected, for a given sequence $\{\rho_k\}$, is that

$$\lim_{k\to\infty} x(\rho_k) = x^*.$$

Differentiation of $P_2(x;\rho)$ gives

$$\nabla P_2(x;\rho) = \nabla f(x) + \sum_{i=1}^{m} \rho \nabla c_i(x) c_i(x) = \nabla f(x) + \rho J(x)^{\mathrm{T}} c(x) \tag{2.10}$$

$$\nabla^2 P_2(x;\rho) = \nabla^2 f(x) + \sum_{i=1}^{m} \rho c_i(x) \nabla^2 c_i(x) + \rho J(x)^{\mathrm{T}} J(x). \tag{2.11}$$

As P_2 is continuously differentiable, its gradient must vanish at the unconstrained minimizer $x(\rho)$. It follows from (2.10) that

$$\nabla f\big(x(\rho)\big) = -\rho J\big(x(\rho)\big)^{\mathrm{T}} c\big(x(\rho)\big),$$

If the constraint qualification holds, this condition has the same form as the first-order necessary condition $\nabla f(x^*) = J(x^*)^{\mathrm{T}} y^*$ (see Theorem 2.2.2). A comparison of these two optimality conditions indicates that the quantity $\pi_i(x;\rho) \triangleq -\rho c_i(x)$, $i = 1, \ldots, m$, may be viewed as an estimate of the i-th Lagrange multiplier at x^*.

The Newton equation to be solved is

$$\nabla^2 P_2(x_k;\rho)\Delta x_k = -\nabla P_2(x_k;\rho).$$

Written in terms of x_k and $\pi(x_k)$, the Newton equation is

$$\big(H(x_k,\pi(x_k)) + \rho J(x_k)^{\mathrm{T}} J(x_k)\big)\Delta x_k = -\big(\nabla f(x_k) - J(x_k)^{\mathrm{T}}\pi(x_k)\big). \tag{2.12}$$

Lemma 2.2.5 (Debreu [7]). *Given an $m \times n$ matrix A and an $n \times n$ symmetric matrix H, then $x^{\mathrm{T}}Hx > 0$ for all nonzero x satisfying $Ax = 0$ if and only if there is a finite $\bar{\rho} \geq 0$ such that $H + \rho A^{\mathrm{T}}A$ is positive definite for all $\rho \geq \bar{\rho}$.*

Debreu's lemma (Lema 2.2.5) implies that if ρ is sufficiently large, these equations are positive definite in the neighborhood of a strict minimizer, i.e., the penalty term adds positive curvature to $H(x^*,y^*)$ in directions orthogonal to null $\big(J(x^*)\big)$.

Algorithm 2.4. Classical Newton Penalty Method.

Fix η_A, γ_c, γ and ε such that $0 < \eta_A < \frac{1}{2}$, $0 < \gamma_c < 1$, $\gamma > 1$, and $0 < \varepsilon \ll 1$;
Choose x_0, ρ ($\rho > 0$);
$x \leftarrow x_0$; $\quad k \leftarrow 0$;
while not *converged* **do**
  while $\|\nabla P_2(x;\rho)\| > \varepsilon$ **do**
    Compute $P_2(x;\rho)$, $\nabla P_2(x;\rho)$;
    Define E such that $\nabla^2 P_2(x;\rho) + E_k$ is positive definite;
    Solve $(\nabla^2 P_2(x;\rho) + E_k)\Delta x = -\nabla P_2(x;\rho)$;
    $\alpha \leftarrow 1$;
    while $P_2(x + \alpha\Delta x;\rho) > P_2(x;\rho) + \eta_A \alpha \nabla P_2(x;\rho)^{\mathrm{T}}\Delta x$ **do**
      $\alpha \leftarrow \gamma_c \alpha$;
    end while
    $x \leftarrow x + \alpha\Delta x$;
    $k \leftarrow k + 1$;
  end while
  $\rho \leftarrow \gamma\rho$;
end while

Augmented Lagrangian Method

The numerical performance of the classical quadratic penalty method becomes poor as $\rho \to \infty$. In order to overcome this difficulty, the augmented Lagrangian method was introduced. This method may be viewed as a shifted penalty function method, which does not require $\mu = 1/\rho$ to go to zero by updating the estimate value of multiplier y in each iteration, and avoid the ill-conditioning. The corresponding problem can be written as

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) - s = 0, \end{aligned} \tag{2.13}$$

where s is a vector of shifts and hence the shifted penalty function is

$$P_2(x;\mu,s) := f(x) + \frac{1}{2}\rho\big(c(x)-s\big)^{\mathrm{T}}\big(c(x)-s\big),$$

and

$$y^E := \rho s$$

may be regarded as an estimate of the Lagrange multiplier.

Expand the shifted penalty function and ignore the constant term $s^{\mathrm{T}}s$, it can be seen minimizing the augmented Lagrangian function is equivalent to minimizing the function below:

$$L_A(x;y,\rho) = f(x) - c(x)^{\mathrm{T}}y + \tfrac{1}{2}\rho c(x)^{\mathrm{T}}c(x), \tag{2.14}$$

The presence of the penalty term in $L_A(x;y^E,\rho)$ has the effect of increasing the (possibly negative) eigenvalues of $H(x^*,y^*)$ corresponding to eigenvectors in the range space of $J(x^*)^{\mathrm{T}}$, while leaving the other eigenvalues unchanged. Using this property, under mild conditions there exists a *finite* $\bar{\rho}$ such that x^* is an *unconstrained minimizer* of $L_A(x;y^*,\rho)$ for all $\rho > \bar{\rho}$. This property is formalized in the following result.

Theorem 2.2.6 (Properties of the augmented Lagrangian (Gill & Wright [17]))**.** *Assume that x^* satisfies the second-order sufficient conditions for a strict minimizer of* (NEP). *Let y^* be*

Lagrange multipliers at x^*. There is a finite $\bar{\rho}$ *such that, for every $\rho > \bar{\rho}$, a solution x^* of* (NEP) *is an isolated local unconstrained minimizer of the augmented Lagrangian function $L_A(x; y^*, \rho)$.*

2.2.4 Optimization Problems with Inequality Constraints

Next we focus on the nonlinear inequality constrained problem (NIP) written in the general form:

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

where $c(x)$ has m components $c_i(x)$, and f and $\{c_i(x)\}$ are smooth functions. The matrix $J(x)$ will denote the Jacobian of the constraint vector $c(x)$. The feasible region for this problem is given by

$$\mathscr{F} = \big\{ x : c_i(x) \geq 0, \;\; i = 1, 2, \ldots, m \big\}.$$

Definition 2.2.14. *The constraint $c_i(x) \geq 0$ is said to be* satisfied *at $\bar{x}$ if $c_i(\bar{x}) \geq 0$,* active *(*binding, satisfied exactly*) if $c_i(\bar{x}) = 0$,* inactive *if $c_i(\bar{x}) > 0$, and* violated *if $c_i(\bar{x}) < 0$.*

Definition 2.2.15 (First-order KKT point for (NIP))**.** *The* first-order KKT conditions *for the inequality-constrained problem* (NIP) *hold at the point x^*, or, equivalently, x^* is a (first-order)* KKT point*, if there exists an m_a-vector y_a^*, called a* Lagrange multiplier vector*, such that*

$$c(x^*) \geq 0, \quad c_a(x^*) = 0, \qquad \text{(feasibility)} \tag{2.15a}$$

$$\nabla f(x^*) = J_a(x^*)^{\mathrm{T}} y_a^*, \qquad \text{(stationarity)} \tag{2.15b}$$

$$y_a^* \geq 0. \qquad \text{(nonnegativity of the multipliers)} \tag{2.15c}$$

Definition 2.2.16 (Acceptable Lagrange multipliers)**.** *Given a KKT point x^* for problem* (NIP)*, the set of* acceptable Lagrange multipliers *is defined as*

$$\mathscr{Y}(x^*) = \big\{ y \in \mathbb{R}^m : \nabla f(x^*) = J(x^*)^{\mathrm{T}} y, \;\; y \geq 0, \;\; \textit{and} \;\; c(x^*) \cdot y = 0 \big\}. \tag{2.16}$$

At any KKT point x, choose some $y \in \mathscr{Y}(x)$ and let $\mathscr{A}_+(x, y)$ denote the set of indices of active constraints with *positive* Lagrange multipliers and let $J_+(x)$ denote the corresponding

matrix of constraint gradients. Similarly, let $\mathscr{A}_0(x,y)$ denote the set of indices of active constraints with zero multipliers, and let $J_0(x)$ denote the associated matrix of constraint gradients.

Theorem 2.2.7 (First-order necessary conditions)**.** *Let x^* be a point such that $c(x^*) \geq 0$, with $c_a(x) = 0$. If the Abadie constraint qualification holds at x^*, then x^* is a local minimizer of* (NIP) *only if x^* is a first-order KKT point, i.e., there exists a vector y_a^* such that*

$$\nabla f(x^*) = J_a(x^*)^{\mathrm{T}} y_a^*, \;\; \textit{with} \;\; y_a^* \geq 0. \tag{2.17}$$

Definition 2.2.17 (Second-order constraint qualification (SOCQ))**.** *The second-order constraint qualification for inequality constraints holds at a KKT point x if, for all $y \in \mathscr{Y}(x)$, every nonzero p satisfying $J_+(x)p = 0$ and $J_0(x)p \geq 0$ is tangent to a twice-differentiable path $x(\alpha)$ such that $c_+\big(x(\alpha)\big) = 0$ and $c_0\big(x(\alpha)\big) \geq 0$ for all $0 < \alpha \leq \widehat{\alpha}$.*

Theorem 2.2.8 (Second-order necessary conditions for (NIP))**.** *If the first- and second-order constraint qualifications hold at x^*, then x^* is a local solution of* (NIP) *only if*

(a) *x^* is a KKT point, i.e., $c(x^*) \geq 0$ and there exists a nonempty set $\mathscr{Y}(x^*)$ of multipliers y satisfying $y \geq 0$, $c(x^*) \cdot y = 0$, and $\nabla f(x^*) = J(x^*)^{\mathrm{T}} y$;*

(b) *for some $y \in \mathscr{Y}(x^*)$ and all $p \neq 0$ satisfying $\nabla f(x^*)^{\mathrm{T}} p = 0$ and $J_a(x^*)p \geq 0$, it holds that $p^{\mathrm{T}} H(x^*, y) p \geq 0$.*

Theorem 2.2.9 (Sufficient conditions for a strict minimizer of (NIP))**.** *A point x^* is a strict local constrained minimizer of* (NIP) *if*

(a) *x^* is a KKT point, i.e., $c(x^*) \geq 0$ and there exists a nonempty set $\mathscr{Y}(x^*)$ of multipliers y satisfying $y \geq 0$, $c(x^*) \cdot y = 0$, and $\nabla f(x^*) = J(x^*)^{\mathrm{T}} y$.*

(b) *There exists a vector $y \in \mathscr{Y}(x^*)$ such that for all $p \neq 0$ satisfying $\nabla f(x^*)^{\mathrm{T}} p = 0$ and $J_a(x^*)p \geq 0$, there is an $\omega > 0$ such that $p^{\mathrm{T}} H(x^*, y) p \geq \omega \|p\|^2$.*

Barrier Methods

For inequality constraints, a *barrier* method is motivated by unconstrained minimization of a function combining f and a positively weighted "barrier" that prevents iterates from leaving the feasible region. *Penalty* methods, in contrast, are based on minimizing a function that includes f and a positive penalty if evaluated at any infeasible point.

The overwhelmingly predominant barrier function used today is the *logarithmic barrier function*:

$$B(x;\mu) = f(x) - \mu \sum_{i=1}^{m} \ln c_i(x), \tag{2.18}$$

Algorithm 2.5. Classical barrier algorithm

```
1:  procedure BARRIER METHOD
2:      Choose x_0 so that c(x_0) > 0. Choose μ > 0, 0 < γ < 1;
3:      k ← 0;
4:      while not converged do
5:          Compute x(μ), an unconstrained minimizer of B(x; μ);
6:          x_{k+1} ← x(μ);
7:          μ ← γμ;
8:          k ← k + 1;
9:      end while
10: end procedure
```

2.2.5 Modified Barrier Function

The unconstrained minimizers of the classical log barrier function converge to a solution of the constrained problem only if the barrier parameter μ goes to zero. By contrast, *modified barrier methods* [3, 5, 18, 26, 28] define a sequence of unconstrained problems in which the value of μ remains *bounded away from zero*, thereby avoiding the need to solve a problem whose Hessian becomes increasingly ill-conditioned as μ is decreased.

Modified barrier methods are based on the observation that for a fixed positive μ, the constraints $c_i(x) \geq 0$ and $\mu \ln\left(1 + c_i(x)/\mu\right) \geq 0$ are equivalent, i.e., their associated sets of feasible points are identical. Moreover, a KKT point for the original problem (NIP) is also a KKT

point for the modified problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \ f(x) \quad \text{subject to} \ \ \mu \ln\left(1+c_i(x)/\mu\right) \geq 0, \ \ i=1,2,\ldots,m. \tag{2.19}$$

This motivates the definition of the *modified barrier function*:

$$M(x,y) = f(x) - \mu \sum_{i=1}^{m} y_i \ln\left(1+c_i(x)/\mu\right), \tag{2.20}$$

which can be interpreted as the conventional Lagrangian function for the modified problem (2.19).

Chapter 3

Computing the Trust-Region Step

This chapter concerns the formulation of methods for solving the trust-regions subproblem. The trust-region subproblem can be written as follows:

$$\underset{d\in\mathbb{R}^n}{\text{minimize}}\ \ \mathscr{Q}(d) = g^{\mathrm{T}}d + \tfrac{1}{2}d^{\mathrm{T}}Hd \quad \text{subject to}\ \ \|d\|_T \leq \delta, \tag{3.1}$$

where $\|d\|_T = (d^{\mathrm{T}}Td)^{1/2}$, δ is the trust-region radius, and T is positive definite. If we write $q = Nd$, where N is the non-singular matrix such that $T = N^{\mathrm{T}}N$, then the problem (3.1) is equivalent to

$$\underset{q\in\mathbb{R}^n}{\text{minimize}}\ \ \hat{\mathscr{Q}}(q) = g^{\mathrm{T}}N^{-1}q + \tfrac{1}{2}q^{\mathrm{T}}N^{-\mathrm{T}}HN^{-1}q \quad \text{subject to}\ \ \|q\| \leq \delta,$$

where $\|\cdot\|$ denotes the two-norm. This problem is just

$$\underset{q\in\mathbb{R}^n}{\text{minimize}}\ \ \widehat{g}^{\mathrm{T}}q + \tfrac{1}{2}q^{\mathrm{T}}\widehat{H}q \quad \text{subject to}\ \ \|q\| \leq \delta, \tag{3.2}$$

with $\widehat{g} = N^{-1}g$ and $\widehat{H} = N^{-\mathrm{T}}HN^{-1}$. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and $\widehat{u}_1$, $\widehat{u}_2$, $\ldots$, $\widehat{u}_n$ denote the eigenvalues and eigenvectors of $\widehat{H}$. If $u_i = N^{-1}\widehat{u}_i$, then $Hu_i = \lambda_i N^{\mathrm{T}}Nu_i$. The least eigenvalue of $\widehat{H} + \sigma I$ satisfies the identity

$$\omega_n = \min_{\substack{y\in\mathbb{R}^n \\ y\neq 0}} \frac{y^{\mathrm{T}}\big(\widehat{H} + \sigma I\big)y}{\|y\|^2} = \min_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \frac{x^{\mathrm{T}}\big(H + \sigma N^{\mathrm{T}}N\big)x}{\|x\|_T^2}. \tag{3.3}$$

In this note, we discuss the details of an iterative method for finding a solution of problem

(3.2). The notation $H(\sigma) = H + \sigma N^{\mathrm{T}} N$ will be used when convenient.

The method we use to find an approximate global solution of the trust-region subproblem is based on an algorithm proposed by Moré and Sorensen [25].

3.1 Solving the Trust-Region Subproblem

Algorithms for finding a global solution of the trust-region subproblem (3.1) are based on the following theorem.

Result 3.1.1. *A vector d^* is a global minimizer of the trust-region subproblem if and only if $\|d^*\|_T \leq \delta$ and there is a $\sigma^* \geq 0$ such that*

$$(H + \sigma^* T)d^* = -g \;\; \textit{and} \;\; \sigma^*(\delta - \|d^*\|_T) = 0, \tag{3.4}$$

with $H + \sigma^* T$ positive semi-definite. Moreover, if $H + \sigma^* T$ is positive definite, then the global minimizer is unique.

If λ_n denotes $\lambda_{\min}(N^{-\mathrm{T}} H N^{-1})$, then $-\lambda_n$ is the unique value such that $H + \sigma N^{\mathrm{T}} N$ is positive semidefinite if $\sigma \geq -\lambda_n$, and singular if $\sigma = -\lambda_n$. For $\sigma > -\lambda_n$ we define the vector

$$s_\sigma = -(H + \sigma N^{\mathrm{T}} N)^{-1} g,$$

which is a unique function of σ. Similarly, we define the scalar-valued function

$$\psi(\sigma) = \|N s_\sigma\| - \delta = \|s_\sigma\|_T - \delta,$$

which is well-defined for all $\sigma > -\lambda_n$. If $\sigma^* \neq 0$ and $\sigma^* \neq -\lambda_n$, then $\psi(\sigma^*) = 0$.

The function $\psi(\sigma)$ may be written as an explicit function of the eigenvalues and eigen-

vectors of $\widehat{H}$. We have

$$\begin{aligned} s_\sigma = -(H+\sigma T)^{-1}g &= -(H+\sigma N^{\mathrm{T}}N)^{-1}g \\ &= -\left(N^{\mathrm{T}}(\widehat{H}+\sigma I)N\right)^{-1}g \\ &= -N^{-1}\left(\widehat{H}+\sigma I\right)^{-1}N^{-\mathrm{T}}g \\ &= -N^{-1}\left(\widehat{U}(\Lambda+\sigma I)^{-1}\widehat{U}^{\mathrm{T}}N^{-\mathrm{T}}g \right. \\ &= -N^{-1}\widehat{U}(\Lambda+\sigma I)^{-1}U^{\mathrm{T}}g, \end{aligned}$$

where U and $\widehat{U}$ are the matrices with columns $\left\{ u_i \right\}$ and $\left\{ \widehat{u}_i \right\}$, respectively. Then

$$Ns_\sigma = -\widehat{U}(\Lambda+\sigma I)^{-1}U^{\mathrm{T}}g = -\sum_{i=1}^{n}\frac{u_i^{\mathrm{T}}g}{\sigma+\lambda_i}\widehat{u}_i = -\sum_{i=1}^{n}\frac{\gamma_i}{\sigma+\lambda_i}\widehat{u}_i, \tag{3.5}$$

where $\gamma_i = u_i^{\mathrm{T}}g$. It follows that $\|Ns_\sigma\|$ is a well-defined function of σ for $\sigma > -\lambda_n$, with

$$\|s_\sigma\|_T^2 = \|Ns_\sigma\|^2 = \frac{\gamma_1^2}{(\lambda_1+\sigma)^2} + \frac{\gamma_2^2}{(\lambda_2+\sigma)^2} + \cdots + \frac{\gamma_n^2}{(\lambda_n+\sigma)^2}. \tag{3.6}$$

3.1.1 A method based on zero-finding.

We have shown that if $\sigma^* \neq 0$ and $\sigma^* \neq -\lambda_n$, then $\psi(\sigma^*) = 0$, and the trust-region subproblem is equivalent to finding a zero of $\psi(\sigma)$. Suppose that $\widehat{\sigma}$ is a zero of $\psi(\sigma)$ such that $H+\widehat{\sigma}T$ is positive semidefinite—i.e., $\psi(\widehat{\sigma}) = 0$ and $\widehat{\sigma} \geq -\lambda_n$. Then $\sigma^* = \max\left\{0, \widehat{\sigma}\right\}$ is the unique value of σ discussed in Result 3.1.1

The following result shows that if $g \neq 0$, the function $\psi(\sigma)$ is strictly decreasing and strictly convex on $(-\lambda_n, \infty)$.

Result 3.1.2. *The function $\psi(\sigma) = \|s_\sigma\|_T - \delta$ is nonincreasing and convex on $(-\lambda_n, \infty)$. Moreover, if $g \neq 0$, then $\psi(\sigma)$ is strictly decreasing and strictly convex on $(-\lambda_n, \infty)$.*

Proof. If $g = 0$, $\psi(\sigma)$ is constant, which is trivially nonincreasing and convex. For the remainder of the proof, we assume that $g \neq 0$. First we show that $\psi'(\sigma) < 0$ for all $\sigma \in (-\lambda_n, \infty)$. For all such σ, the matrix $H+\sigma T$ is positive definite and the vector s_σ is well-defined with $\|s_\sigma\|_T > 0$.

Differentiating $\psi(\sigma) = \|s_\sigma\|_T - \delta$ with respect to σ gives

$$\psi'(\sigma) = \frac{d}{d\sigma}\big(\|s_\sigma\|_T\big). \tag{3.7}$$

The derivative of $\|s_\sigma\|_T$ is given by

$$\begin{aligned}\frac{d}{d\sigma}(\|s_\sigma\|_T) &= \frac{1}{2}\frac{1}{\|s_\sigma\|_T}\frac{d}{d\sigma}\big(s_\sigma^{\mathrm{T}} T s_\sigma\big) \\ &= \frac{1}{2}\frac{1}{\|s_\sigma\|_T}\left(\Big(\frac{d}{d\sigma}s_\sigma\Big)^{\mathrm{T}} T s_\sigma + s_\sigma^{\mathrm{T}} T \frac{d}{d\sigma}s_\sigma\right) \\ &= \frac{1}{\|s_\sigma\|_T} w_\sigma^{\mathrm{T}} T s_\sigma, \end{aligned} \tag{3.8}$$

where $w_\sigma = ds_\sigma/d\sigma$. In order to obtain an expression for w_σ, we differentiate the equation $(H+\sigma T)s_\sigma = -g$ with respect to σ to obtain

$$(H+\sigma T)\frac{d}{d\sigma}s_\sigma + T s_\sigma = 0,$$

in which case

$$(H+\sigma T)w_\sigma = -T s_\sigma, \quad \text{i.e.,} \quad w_\sigma = -(H+\sigma T)^{-1} T s_\sigma. \tag{3.9}$$

Forming the inner product of w_σ from (3.9) with Ts_σ gives $w_\sigma^{\mathrm{T}} T s_\sigma = -s_\sigma^{\mathrm{T}} T (H+\sigma T)^{-1} T s_\sigma$, and $\psi'(\sigma)$ may be written as

$$\psi'(\sigma) = -\frac{s_\sigma^{\mathrm{T}} T (H+\sigma T)^{-1} T s_\sigma}{\|s_\sigma\|_T}. \tag{3.10}$$

An alternative form for $\psi'(\sigma)$ may be determined by forming the inner product of w_σ with the first expression of (3.9) to give $w_\sigma^{\mathrm{T}} T s_\sigma = -w_\sigma^{\mathrm{T}}(H+\sigma T)w_\sigma$. It follows from (3.7) and (3.8) that $\psi'(\sigma)$ may be written as

$$\psi'(\sigma) = -\frac{w_\sigma^{\mathrm{T}}(H+\sigma T)w_\sigma}{\|s_\sigma\|_T}. \tag{3.11}$$

As $H+\sigma T$ is positive definite for every $\sigma \in (-\lambda_n, \infty)$, it follows that $\psi'(\sigma) < 0$ and ψ is strictly decreasing for all $\sigma \in (-\lambda_n, \infty)$.

For the second derivative we differentiate $\psi'(\sigma)$ stated in the form (3.8) and use (3.9) to

give

$$\begin{aligned}\psi''(\sigma) &= -\frac{1}{\|s_\sigma\|_T^2}(w_\sigma^{\mathrm{T}} T s_\sigma)\frac{d}{d\sigma}(\|s_\sigma\|_T) + \frac{1}{\|s_\sigma\|_T}\left(s_\sigma^{\mathrm{T}} T\left(\frac{dw_\sigma}{d\sigma}\right) + w_\sigma^{\mathrm{T}} T w_\sigma\right) \\ &= -\frac{1}{\|s_\sigma\|_T^3}(w_\sigma^{\mathrm{T}} T s_\sigma)^2 + \frac{1}{\|s_\sigma\|_T}\left(s_\sigma^{\mathrm{T}} T\left(\frac{dw_\sigma}{d\sigma}\right) + w_\sigma^{\mathrm{T}} T w_\sigma\right). \end{aligned} \tag{3.12}$$

Similarly, differentiating the identity $(H+\sigma T)w_\sigma = -Ts_\sigma$ from (3.9) with respect to σ gives

$$(H+\sigma T)\frac{dw_\sigma}{d\sigma} + Tw_\sigma + T\frac{d}{d\sigma}s_\sigma = (H+\sigma T)\frac{dw_\sigma}{d\sigma} + 2Tw_\sigma = 0.$$

Premultiplying by w_σ^{T} and using the expression $w_\sigma = -(H+\sigma T)^{-1}Ts_\sigma$ from (3.9) yields

$$w_\sigma^{\mathrm{T}}(H+\sigma T)\frac{dw_\sigma}{d\sigma} + 2w_\sigma^{\mathrm{T}} T w_\sigma = 0, \text{ or, equivalently, } s_\sigma^{\mathrm{T}} T\left(\frac{dw_\sigma}{d\sigma}\right) = 2w_\sigma^{\mathrm{T}} T w_\sigma.$$

If this expression is substituted in (3.12) we get

$$\begin{aligned}\psi''(\sigma) &= \frac{3w_\sigma^{\mathrm{T}} T w_\sigma}{\|s_\sigma\|_T} - \frac{(w_\sigma^{\mathrm{T}} T s_\sigma)^2}{\|s_\sigma\|_T^3} \\ &= \frac{1}{\|s_\sigma\|_T^3}\left(2(w_\sigma^{\mathrm{T}} T w_\sigma)(s_\sigma^{\mathrm{T}} T s_\sigma) + (w_\sigma^{\mathrm{T}} T w_\sigma)(s_\sigma^{\mathrm{T}} T s_\sigma) - (w_\sigma^{\mathrm{T}} T s_\sigma)^2\right). \end{aligned} \tag{3.13}$$

From (3.6) and the assumption that $g \neq 0$, it must hold that both $\|s_\sigma\|_T$ and $\|w_\sigma\|_T$ are nonzero. Moreover, the Cauchy-Schwartz inequality implies that $(w_\sigma^{\mathrm{T}} T w_\sigma)(s_\sigma^{\mathrm{T}} T s_\sigma) - (w_\sigma^{\mathrm{T}} T s_\sigma)^2 \geq 0$. It follows that $\psi''(\sigma)$ is positive and hence $\psi(\sigma)$ is strictly convex for all $\sigma \in (-\lambda_n, \infty)$. □

The next result gives conditions under which the existence of a zero $\widehat{\sigma}$ of $\psi(\sigma)$ in $(-\lambda_n, \infty)$ is guaranteed.

Result 3.1.3. *If* $\lim_{\sigma \to -\lambda_n} \|s_\sigma\|_T > \delta$ *then* $\psi(\sigma)$ *has a unique zero in* $(-\lambda_n, \infty)$.

Proof. The expansion (3.6) implies that $\lim_{\sigma\to\infty} \|s_\sigma\|_T = 0$ and hence $\lim_{\sigma\to\infty} \psi(\sigma) = -\delta < 0$. If $\lim_{\sigma \to -\lambda_n} \|s_\sigma\|_T > \delta$, then $\lim_{\sigma \to -\lambda_n} \psi(\sigma) > 0$ and it follows that $\psi(\sigma)$ changes sign on $(-\lambda_n, \infty)$. Moreover, from Result 3.1.2, $\psi(\sigma)$ is convex and therefore continuous on $(-\lambda_n, \infty)$. The intermediate-value theorem then implies that $\psi(\sigma)$ must have at least one zero in $(-\lambda_n, \infty)$. The uniqueness now follows from the fact that $\psi(\sigma)$ is strictly decreasing on $(-\lambda_n, \infty)$. □

The zero of $\psi(\sigma)$ may be found using an appropriate method for one-dimensional zero-finding. The usual strategy is to attempt to solve $\psi(\sigma) = 0$ for $\widehat{\sigma}$ using some variant of Newton's method, and set $\sigma = 0$ during the computation if it appears that the iterates are converging to a negative value of $\widehat{\sigma}$. Suppose that $\sigma_j \in (-\lambda_n, \infty)$ is the jth estimate of $\widehat{\sigma}$. Newton's method for finding a zero of $\psi(\sigma)$ defines the new estimate $\sigma_{j+1}^N = \sigma_j - \psi(\sigma_j)/\psi'(\sigma_j)$, which is the zero of the affine model $\psi(\sigma_j) + \psi'(\sigma_j)(\sigma - \sigma_j)$. Note that σ_{j+1}^N is well-defined because $\psi'(\sigma_j) \neq 0$ from Result 3.1.2. Newton's method has special properties when used to find the zero of a strictly decreasing convex function. These properties are summarized in the following result.

Result 3.1.4. *Suppose that there exists a $\widehat{\sigma} \in (-\lambda_n, \infty)$ such that $\psi(\widehat{\sigma}) = 0$. Let $\sigma_{j+1}^N = \sigma_j - \psi(\sigma_j)/\psi'(\sigma_j)$ denote the Newton iterate defined at some $\sigma_j \in (-\lambda_n, \infty)$. Then*

(a) *if $\sigma_j \neq \widehat{\sigma}$, the product $\psi(\sigma_j)(\sigma_j - \widehat{\sigma})$ is negative;*

(b) *if $\sigma_j > \widehat{\sigma}$ then $\sigma_{j+1}^N < \widehat{\sigma}$;*

(c) *if $\sigma_j \in (-\lambda_n, \widehat{\sigma})$ then $\sigma_{j+1}^N > \sigma_j$ and $\sigma_{j+1}^N \in (-\lambda_n, \widehat{\sigma})$; and*

(d) *if $\sigma_j \in (-\lambda_n, \widehat{\sigma})$ then all subsequent Newton iterates increase monotonically and converge to $\widehat{\sigma}$.* ∎

A safeguarded Newton method based on Result 3.1.4 is given in Algorithm 3.1.

Algorithm 3.1. Safeguarded Newton method.

Choose $\sigma_0 \geq -\lambda_n$;
converged $\leftarrow$ false; $j \leftarrow 0$;
while not *converged* **do**
  $\sigma_{j+1}^N \leftarrow \sigma_j - \psi(\sigma_j)/\psi'(\sigma_j)$; [$\sigma_{j+1}^N < \widehat{\sigma}$, but σ_{j+1}^N may not lie in $(-\lambda_n, \infty)$.]
  $\sigma_{j+1} \leftarrow$ **if** $\sigma_{j+1}^N > -\lambda_n$ **then** σ_{j+1}^N **else** $\frac{1}{2}(\sigma_j - \lambda_n)$;
  $j \leftarrow j + 1$;
end while

Improving the efficiency of Newton's method for finding $\widehat{\sigma}$.

Unfortunately, the nonlinear equation $\psi(\sigma) = 0$ cannot be solved efficiently using Newton's method. Implicitly, an iteration of Newton's method involves finding the zero of a linear model of ψ at σ_j^N. The expression (3.6) implies that $\psi(\sigma)$ has a pole at $-\lambda_n$, and so $\psi(\sigma)$ cannot be approximated very accurately by a linear function near $-\lambda_n$. Hebden [22] avoids this difficulty by using a local model of $\psi(\sigma)$ that also has a pole but provides a better approximation of $\psi(\sigma)$. An equivalent approach is to search for a zero of the function

$$\varphi(\sigma) = \frac{1}{\delta} - \frac{1}{\|s_\sigma\|_T}$$

(see Reinsch [30]). This function has no poles and has the same roots as ψ. Note that φ is not differentiable at $\sigma = -\lambda_n$.

Result 3.1.5. *The function $\varphi'(\sigma)$ is nonincreasing and convex on $(-\lambda_n, \infty)$. Moreover, if $g \neq 0$, then $\varphi(\sigma)$ is strictly decreasing and strictly convex on $(-\lambda_n, \infty)$.*

Proof. First, we derive $\varphi'(\sigma)$ and $\varphi''(\sigma)$. Assume that $H + \sigma T$ is nonsingular. Differentiating $\varphi(\sigma)$ with respect to σ gives

$$\varphi'(\sigma) = -\frac{d}{d\sigma}\big(\|s_\sigma\|_T^{-1}\big) = \frac{1}{\|s_\sigma\|_T^2}\frac{d}{d\sigma}\big(\|s_\sigma\|_T\big). \tag{3.14}$$

From (3.8) and (3.9), the derivative of $\|s_\sigma\|_T$ is given by

$$\frac{d}{d\sigma}(\|s_\sigma\|_T) = \frac{1}{\|s_\sigma\|_T} w_\sigma^{\mathrm{T}} T s_\sigma, \text{ where } w_\sigma = \frac{d}{d\sigma} s_\sigma = -(H + \sigma T)^{-1} T s_\sigma. \tag{3.15}$$

It follows from this last identity and (3.14) that

$$\varphi'(\sigma) = \frac{1}{\|s_\sigma\|_T^3} w_\sigma^{\mathrm{T}} T s_\sigma. \tag{3.16}$$

Forming the inner-product of w_σ and Ts_σ gives $w_\sigma^{\mathrm{T}} T s_\sigma = -s_\sigma^{\mathrm{T}} T (H + \sigma T)^{-1} T s_\sigma$, which may be used to write (3.16) as

$$\varphi'(\sigma) = -\frac{s_\sigma^{\mathrm{T}} T (H + \sigma T)^{-1} T s_\sigma}{\|s_\sigma\|_T^3}. \tag{3.17}$$

An alternative form for $\varphi'(\sigma)$ may be determined by forming the inner product of w_σ with the first expression of (3.9) to give $w_\sigma^{\mathrm{T}} T s_\sigma = -w_\sigma^{\mathrm{T}}(H+\sigma T)w_\sigma$. It follows that (3.16) may be written as

$$\varphi'(\sigma) = -\frac{w_\sigma^{\mathrm{T}}(H+\sigma T)w_\sigma}{\|s_\sigma\|_T^3}. \tag{3.18}$$

In order to show that φ is strictly convex we show that $\varphi''(\sigma)$ is strictly positive for all $\sigma > -\lambda_n$. For the second derivative we differentiate $\varphi'(\sigma)$ stated in the form (3.16) and use (3.15) to give

$$\begin{aligned}\varphi''(\sigma) &= -\frac{3}{\|s_\sigma\|_T^4}(w_\sigma^{\mathrm{T}} T s_\sigma)\frac{d}{d\sigma}(\|s_\sigma\|_T) + \frac{1}{\|s_\sigma\|_T^3}\left(s_\sigma^{\mathrm{T}} T\left(\frac{dw_\sigma}{d\sigma}\right) + w_\sigma^{\mathrm{T}} T w_\sigma\right) \\ &= -\frac{3}{\|s_\sigma\|_T^5}(w_\sigma^{\mathrm{T}} T s_\sigma)^2 + \frac{1}{\|s_\sigma\|_T^3}\left(s_\sigma^{\mathrm{T}} T\left(\frac{dw_\sigma}{d\sigma}\right) + w_\sigma^{\mathrm{T}} T w_\sigma\right).\end{aligned} \tag{3.19}$$

Similarly, differentiating the identity $(H+\sigma T)w_\sigma = -T s_\sigma$ from (3.9) with respect to σ gives

$$(H+\sigma T)\frac{dw_\sigma}{d\sigma} + T w_\sigma + T\frac{d}{d\sigma}s_\sigma = (H+\sigma T)\frac{dw_\sigma}{d\sigma} + 2T w_\sigma = 0.$$

Premultiplying by w_σ^{T} and using the expression $w_\sigma = -(H+\sigma T)^{-1} T s_\sigma$ from (3.9) yields

$$w_\sigma^{\mathrm{T}}(H+\sigma T)\frac{dw_\sigma}{d\sigma} + 2w_\sigma^{\mathrm{T}} T w_\sigma = 0, \text{ or, equivalently, } s_\sigma^{\mathrm{T}} T\left(\frac{dw_\sigma}{d\sigma}\right) = 2w_\sigma^{\mathrm{T}} T w_\sigma.$$

If this expression is substituted in (3.19), φ'' may be rearranged so that

$$\varphi''(\sigma) = \frac{3}{\|s_\sigma\|_T^5}\Big((w_\sigma^{\mathrm{T}} T w_\sigma)(s_\sigma^{\mathrm{T}} T s_\sigma) - (w_\sigma^{\mathrm{T}} T s_\sigma)^2\Big). \tag{3.20}$$

From the definition of s_σ the expression (3.6) gives

$$\|s_\sigma\|_T^2 = \|(H+\sigma N^{\mathrm{T}} N)^{-1} g\|_T^2 = \sum_{i=1}^{n} \frac{(u_i^{\mathrm{T}} g)^2}{(\sigma+\lambda_i)^2}.$$

If $u_n^{\mathrm{T}} g \neq 0$, the definition of $\varphi(\sigma)$ implies that

$$\lim_{\sigma\to-\lambda_n} \varphi(\sigma) = \frac{1}{\delta}, \qquad \lim_{\sigma\to+\infty} \varphi(\sigma) = -\infty$$

and the intermediate-value theorem implies that there exists at least one zero of $\varphi(\sigma)$ in $(-\lambda_n, \infty)$.

From (3.18), the first derivative of $\varphi(\sigma)$ is

$$\varphi'(\sigma) = -\frac{w_\sigma^{\mathrm{T}}(H+\sigma T)w_\sigma}{\|s_\sigma\|_T^3},$$

which is strictly negative for all $\sigma \in (-\lambda_n, \infty)$. It follows that $\varphi(\sigma)$ is strictly decreasing in $(-\lambda_n, \infty)$. Similarly, the second derivative of $\varphi(\sigma)$ is given by

$$\varphi''(\sigma) = \frac{3}{\|s_\sigma\|_T^5}\Big((w_\sigma^{\mathrm{T}} T w_\sigma)(s_\sigma^{\mathrm{T}} T s_\sigma) - (w_\sigma^{\mathrm{T}} T s_\sigma)^2\Big)$$

(see (3.20)). The Cauchy-Schwartz inequality implies

$$(w_\sigma^{\mathrm{T}} T w_\sigma)(s_\sigma^{\mathrm{T}} T s_\sigma) - (w_\sigma^{\mathrm{T}} T s_\sigma)^2 \geq 0,$$

with equality only if Nw_σ is a multiple of Ns_σ. If Nw_σ were a multiple of Ns_σ, equation (3.9) would imply that Ns_σ is an eigenvector of $\widehat{H} + \sigma I$ and hence a multiple of $N^{-\mathrm{T}}g$. This would imply that $u_n^{\mathrm{T}} g = 0$, which violates the assumption that $u_n^{\mathrm{T}} g \neq 0$. Hence $\varphi''(\sigma)$ is strictly positive for all $\sigma > -\lambda_n$. It follows that φ is strictly decreasing and strictly convex in $(-\lambda_n, \infty)$. □

Result 3.1.6. *If $u_n^{\mathrm{T}} g \neq 0$, then φ has a unique zero $\widehat{\sigma}$ in $(-\lambda_n, \infty)$.*

Proof. We have shown that φ is strictly decreasing and strictly convex in $(-\lambda_n, \infty)$, i.e., $\varphi'(\sigma) < 0$ and $\varphi''(\sigma) > 0$ for all $\sigma \in (-\lambda_n, \infty)$. It follows that $\varphi(\sigma)$ must have a unique zero in $(-\lambda_n, \infty)$. □

The equation $\varphi(\sigma) = 0$ is solved using Newton's method. The Newton step is given by $\sigma_{j+1} = \sigma_j - \varphi(\sigma_j)/\varphi'(\sigma_j)$. The work is all in evaluating $\varphi'(\sigma)$. As in the proof of Result 3.1.5, differentiating $\varphi(\sigma)$ with respect to σ gives

$$\varphi'(\sigma) = -\frac{w^{\mathrm{T}}(H+\sigma N^{\mathrm{T}} N)w}{\|Ns_\sigma\|^3}.$$

Substituting for φ and φ' in the Newton equation gives

$$\begin{aligned}\sigma_{j+1} &= \sigma_j + \frac{\|Ns_j\|^2}{w_j^{\mathrm{T}}(H+\sigma_j N^{\mathrm{T}}N)w_j}\left(\frac{\|Ns_j\|-\delta}{\delta}\right)\\ &= \sigma_j + \frac{\|Ns_j\|^2}{s_j^{\mathrm{T}}N^{\mathrm{T}}N(H+\sigma_j N^{\mathrm{T}}N)^{-1}N^{\mathrm{T}}Ns_j}\left(\frac{\|Ns_j\|-\delta}{\delta}\right)\\ &= \sigma_j + \frac{\|s_j\|_T^2}{s_j^{\mathrm{T}}T(H+\sigma_j T)^{-1}Ts_j}\left(\frac{\|s_j\|_T-\delta}{\delta}\right),\end{aligned}$$

as required.

If there is any value of $\sigma > -\lambda_n$ for which $\|Ns_\sigma\| = \delta$, then that value is unique. Clearly then, the solution of $\psi(\sigma) = 0$ in the interval $[-\lambda_n, \infty]$ is also unique if it exists. Let this value be $\bar{\sigma}$. If $\bar{\sigma} > 0$, then $\sigma^* = \bar{\sigma}$. If $\bar{\sigma} < 0$, then $\sigma^* = 0$. If $\bar{\sigma}$ does not exist, then either $\sigma^* = 0$ or $H(-\lambda_n)s = -g$ is compatible and $\sigma^* = -\lambda_n$. It will be shown that a Newton iteration will always detect $\sigma^* = 0$, whether or not $\bar{\sigma}$ exists, but if $\sigma^* = -\lambda_n$, additional techniques must be employed to find a solution to the trust-region subproblem.

3.1.2 The degenerate case.

If the least eigenvalue of H is distinct, i.e., $\lambda_{n-1} > \lambda_n$, the expansion (3.6) implies that if $u_n^{\mathrm{T}}g \neq 0$, then $\lim_{\sigma\to-\lambda_n}\|s_\sigma\|_T = \infty$, and hence, from Result 3.1.3, that $\psi(\sigma)$ has a unique zero in $(-\lambda_n, \infty)$. If $u_n^{\mathrm{T}}g = 0$, then u_n is orthogonal to the gradient and $\|s_\sigma\|_T$ has no pole at $\sigma = -\lambda_n$, which implies that $\|s_\sigma\|_T$ approaches a finite value $s^\dagger$ as $\sigma \to -\lambda_n$.

The properties of $\psi(\sigma)$ and $\|s_\sigma\|_T$ as $\sigma \to -\lambda_n$ depend on the relationship between g and the invariant subspace

$$\mathscr{S}_n = \left\{ u : Hu = \lambda_n Tu \right\} = \mathrm{null}(H - \lambda_n T)$$

associated with the linear matrix pencil $H - \lambda T$.

Result 3.1.7. *The quantity $\|s_\sigma\|_T = \|(H+\sigma T)^{-1}g\|_T$ is finite as $\sigma \to -\lambda_n$ if and only if g lies in $\mathscr{S}_n^{\perp}$.*

Proof. If $\|s_\sigma\|_T$ is finite as $\sigma \to -\lambda_n$ then the definition (3.6) of $\|s_\sigma\|_T$ in terms of the spectral decomposition of H implies that $u_i^{\mathrm{T}} g = 0$ for all $u_i \in \mathscr{S}_n$. As these u_i form a basis for $\mathscr{S}_n$, it must hold that every $u \in \mathscr{S}_n$ must satisfy $u^{\mathrm{T}} g = 0$. Conversely, if $\|s_\sigma\|_T \to \infty$ as $\sigma \to -\lambda_n$ then (3.6) implies that $u_i^{\mathrm{T}} g \neq 0$ for at least one $u_i \in \mathscr{S}_n$. □

If $\|s_\sigma\|_T$ is finite as $\sigma \to -\lambda_n$, then the trust-region subproblem is said to be *degenerate*.

Result 3.1.8. *Let* $\psi(\sigma) = \|(H+\sigma T)^{-1} g\|_T - \delta = \|s_\sigma\|_T - \delta$.

(a) *The quantity* $\lim_{\sigma \to -\lambda_n} \psi(\sigma)$ *is finite if and only if the linear equations* $(H - \lambda_n T)d = -g$ *are compatible.*

(b) *Moreover, if* $\lim_{\sigma \to -\lambda_n} \psi(\sigma)$ *is finite, then* $\lim_{\sigma \to -\lambda_n} s_\sigma$ *exists and is given by the vector* $s^\dagger = -(H - \lambda_n T)^\dagger g$, *where* $(H - \lambda_n T)^\dagger$ *denotes the pseudoinverse of* $H - \lambda_n T$.

Proof. For part **(a)**, if $\psi(\sigma)$ is finite as $\sigma \to -\lambda_n$, then Result 3.1.7 implies that $u_i^{\mathrm{T}} g = 0$ for all $u_i \in \mathscr{S}_n$. As $\mathscr{S}_n$ is just the null space of $H - \lambda_n T$, it follows that

$$g \in \mathrm{null}(H - \lambda_n T)^\perp = \mathrm{range}(H - \lambda_n T),$$

and the system $(H - \lambda_n T)d = -g$ must be compatible. Conversely, if $\psi(\sigma) \to \infty$ as $\sigma \to -\lambda_n$, then Result 3.1.7 implies that there must exist a $u \in \mathscr{S}_n$ such that $u^{\mathrm{T}} g \neq 0$. The vector g can be expressed uniquely as $g = g\mathbb{R} + g\mathbb{N}$, where $g\mathbb{N} \in \mathrm{null}(H - \lambda_n T)$ and $g\mathbb{R} \in \mathrm{range}(H - \lambda_n T)$. Clearly, $u^{\mathrm{T}} g = u^{\mathrm{T}} g\mathbb{R} + u^{\mathrm{T}} g\mathbb{N} = u^{\mathrm{T}} g\mathbb{N} \neq 0$, which implies that $g\mathbb{N} \neq 0$. If $g\mathbb{N}$ is nonzero, then $g \neq g\mathbb{R}$ and $g \notin \mathrm{range}(H - \lambda_n T)$.

For part **(b)** assume that λ_n, the least eigenvalue of $N^{-\mathrm{T}} H N^{-1}$, has multiplicity $n - r$. If $N^{-\mathrm{T}} H N^{-1} = \widehat{U} \Lambda \widehat{U}^{\mathrm{T}}$, then, for every $\sigma > -\lambda_n$, the matrix $H + \sigma T$ is nonsingular, with

$$s_\sigma = -(H + \sigma T)^{-1} g = -N^{-1} \widehat{U} (\Lambda + \sigma I)^{-1} \widehat{U}^{\mathrm{T}} N^{-\mathrm{T}} g.$$

Consider the following partitions of Λ and $\widehat{U}$

$$\Lambda = \begin{pmatrix} \Lambda_r & 0 \\ 0 & \Lambda_{n-r} \end{pmatrix} = \begin{pmatrix} \Lambda_r & 0 \\ 0 & \lambda_n I_{n-r} \end{pmatrix} \text{ and } \widehat{U} = \big(\underbrace{\widehat{U}_r}_{n\times r} \quad \underbrace{\widehat{U}_{n-r}}_{n\times(n-r)} \big).$$

It follows that

$$\begin{aligned} H - \lambda_n T &= N^{\mathrm{T}} \widehat{U} \begin{pmatrix} \Lambda_r - \lambda_n I_r & 0 \\ 0 & 0 \end{pmatrix} (N^{\mathrm{T}} \widehat{U})^{\mathrm{T}} \\ &= \begin{pmatrix} N^{\mathrm{T}} \widehat{U}_r & N^{\mathrm{T}} \widehat{U}_{n-r} \end{pmatrix} \begin{pmatrix} \Lambda_r - \lambda_n I_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \widehat{U}_r^{\mathrm{T}} N \\ \widehat{U}_{n-r}^{\mathrm{T}} N \end{pmatrix}, \end{aligned}$$

in which case $N^{\mathrm{T}}\widehat{U}_r$ and $N^{\mathrm{T}}\widehat{U}_{n-r}$ define bases for range$(H-\lambda_n T)$ and null$(H-\lambda_n T)$. If $\psi(\sigma)$ is finite as $\sigma \to -\lambda_n$, then part **(a)** implies that $g \in \text{range}(H-\lambda_n T)$, and so $\widehat{U}_{n-r}^{\mathrm{T}} N^{-\mathrm{T}} g = 0$. Then

$$\begin{aligned} s_\sigma &= -N^{-1}\widehat{U}(\Lambda + \sigma I)^{-1}\widehat{U}^{\mathrm{T}} N^{-\mathrm{T}} g \\ &= -\begin{pmatrix} N^{-1}\widehat{U}_r & N^{-1}\widehat{U}_{n-r} \end{pmatrix} \begin{pmatrix} \Lambda_r + \sigma I_r & 0 \\ 0 & (\lambda_n + \sigma) I_{n-r} \end{pmatrix}^{-1} \begin{pmatrix} \widehat{U}_r^{\mathrm{T}} N^{-\mathrm{T}} g \\ \widehat{U}_{n-r}^{\mathrm{T}} N^{-\mathrm{T}} g \end{pmatrix} \\ &= -N^{-1}\widehat{U}_r(\Lambda_r + \sigma I_r)^{-1}\widehat{U}_r^{\mathrm{T}} N^{-\mathrm{T}} g - \frac{1}{\sigma + \lambda_n} N^{-1}\widehat{U}_{n-r}\widehat{U}_{n-r}^{\mathrm{T}} N^{-\mathrm{T}} g \\ &= -N^{-1}\widehat{U}_r(\Lambda_r + \sigma I_r)^{-1}\widehat{U}_r^{\mathrm{T}} N^{-\mathrm{T}} g. \end{aligned}$$

As the matrix $\Lambda_r - \lambda_n I_r$ is nonsingular, we may take the limit as $\sigma \to -\lambda_n$ to give

$$\begin{aligned} s^\dagger &= -N^{-1}\widehat{U}_r(\Lambda_r - \lambda_n I_r)^{-1}\widehat{U}_r^{\mathrm{T}} N^{-\mathrm{T}} g \\ &= -N^{-1}\widehat{U} \begin{pmatrix} (\Lambda_r - \lambda_n I_r)^{-1} & 0 \\ 0 & 0 \end{pmatrix} \widehat{U}^{\mathrm{T}} N^{-\mathrm{T}} g \\ &= -(H - \lambda_n T)^\dagger g, \end{aligned}$$

as required. □

If $g \in \mathscr{S}_n^{\perp}$ and $\|s^{\dagger}\|_T < \delta$, then $\psi(\sigma)$ has no zero in $[-\lambda_n, \infty)$. This situation is often referred to as the *"hard case"*.

In the hard case, two situations are possible. If λ_n is positive, the quantities $\sigma^* = 0$ and $d^* = -H^{-1}g$ satisfy the optimality condition (3.4) because

$$\begin{aligned} \|d^*\|_T &= \|N^{-1}\widehat{U}_r \Lambda_r^{-1} \widehat{U}_r^{\mathrm{T}} N^{-\mathrm{T}} g\|_T \\ &\leq \|N^{-1}\widehat{U}_r (\Lambda_r - \lambda_n I_r)^{-1} \widehat{U}_r^{\mathrm{T}} N^{-\mathrm{T}} g\|_T = \|s^{\dagger}\|_T < \delta. \end{aligned}$$

On the other hand, if λ_n is negative or zero, the system $(H - \lambda_n T)d = -g$ is compatible but cannot be used alone to determine d^*. However, as $H - \lambda_n T$ is singular, there is a null vector z of $H - \lambda_n T$ of unit length and a scalar τ for which

$$(H - \lambda_n T)(s^{\dagger} + \tau z) = -g \quad \text{and} \quad \|s^{\dagger} + \tau z\|_T = \delta.$$

Then $\sigma^* = -\lambda_n$ and $d^* = s^{\dagger} + \tau z$ satisfy optimality conditions (3.4) and hence define a global minimizer of the trust-region subproblem. Any null vector for $H - \lambda_n T$ is also a null vector for $(H - \lambda_n T)^{\dagger}$, and hence $z^{\mathrm{T}} s^{\dagger} = -z^{\mathrm{T}}(H - \lambda_n T)^{\dagger} g = 0$. This identity can be used to provide a simple expression for τ. From the definition of d^* we have

$$\|d^*\|_T^2 = \|s^{\dagger} + \tau z\|_T^2 = \|s^{\dagger}\|_T^2 + \tau^2 \|z\|_T^2 = \|s^{\dagger}\|_T^2 + \tau^2 = \delta^2,$$

which fixes $\tau = \pm(\delta^2 - \|s^{\dagger}\|_T^2)^{1/2}$. In this situation, the trust-region step is not unique because both $d^* = s^{\dagger} + |\tau| z$ and $d^* = s^{\dagger} - |\tau| z$ satisfy (3.4) and define the same global minimum of the quadratic model. In two dimensions, the direction of z is unique and there are just two solutions in the hard case.

3.2 The Method of Moré and Sorensen

The method of Moré and Sorensen [25] is widely regarded as the "standard" method for solving the trust-region subproblem in unconstrained optimization. A trust-region subproblem

may be regarded as having three distinct properties.

- There does or does not exist a $\widehat{\sigma} \geq -\lambda_n$ such that $\psi(\widehat{\sigma}) = 0$.
- The optimal value σ^* does or does not equal zero.
- The equations $(H - \lambda_n T)s = -g$ are either compatible or incompatible.

The Moré-Sorensen algorithm may be considered as having three distinct parts, corresponding to the three properties listed above:

- A safeguarded Newton iteration attempts to find $\widehat{\sigma} \geq -\lambda_n$ for which $\psi(\widehat{\sigma}) = 0$.
- If the current Newton iterate is σ_j, a test is performed on σ_j to keep $\sigma_j \geq 0$ and terminate the algorithm if $\sigma^* = 0$.
- If the current Newton iterate is σ_j and s_j is a solution of the equations $(H + \sigma_j T)s_j = -g$, then an alternative method is used to find an acceptable step if $\|s_j\|_T < \delta$. This method is most likely to succeed if the equations $(H - \lambda_n T)s_j = -g$ are nearly compatible and σ_j is near $-\lambda_n$.

3.2.1 Approximate solution of the trust-region subproblem.

In practice, it is not possible to compute an exact solution of the trust-region subproblem. There are two reasons for this. First, methods for finding a point $\widehat{\sigma}$ such that $\psi(\widehat{\sigma}) = 0$ generate an infinite sequence that must be terminated after a finite number steps. The termination point will not be an exact zero of the equations and may not satisfy the trust-region constraint exactly. Second, the solution of the subproblem in the hard case requires the calculation of a vector in the invariant subspace (i.e., eigenspace) associated with λ_n, the least eigenvalue of H. However, λ_n cannot be computed explicitly and it is not possible to define z_j so that $(H - \lambda_n T)z_j = 0$.

This means that any practical procedure must compute a vector d that approximates solution of the trust-region subproblem. In this section, procedures will be formulated that

provide an approximate global minimizer d in the sense that $\mathcal{Q}(d) \leq \tau \mathcal{Q}(d^*)$ for some $0 < \tau < 1$. However, it must be emphasized that d will not, in general, satisfy the optimality conditions for d^* with high accuracy.

In order to simplify the discussion we will consider each source of error in d separately. First, methods for finding a zero of ψ or φ generate an infinite sequence $\{\sigma_j\}$ that is terminated when $|\psi|$ is smaller than some preassigned tolerance. Unfortunately, as the final value of ψ will not be exactly zero, the resulting estimate of σ may violate the trust-region bound by an amount that depends on the tolerance. To allow for this, we find the approximate zero of ψ defined with a scalar $\tilde{\delta}$ that is a slightly smaller than the trust-region radius δ. Given some tolerance ε such that $0 < \varepsilon < 1$, suppose that the safeguarded Newton iteration is terminated when σ satisfies

$$|\widetilde{\psi}(\sigma)| \leq \varepsilon, \quad \text{where} \quad \widetilde{\psi}(\sigma) = \frac{\|s_\sigma\|_T - \tilde{\delta}}{\tilde{\delta}}, \quad \text{with} \quad (H + \sigma T)s_\sigma = -g. \tag{3.21}$$

Then it must hold that $\|s_\sigma\|_T$ satisfies $(1-\varepsilon)\tilde{\delta} \leq \|s_\sigma\|_T \leq (1+\varepsilon)\tilde{\delta}$. If $\tilde{\delta}$ is chosen such that $\delta = (1+\varepsilon)\tilde{\delta}$, then s_σ satisfies $\|s_\sigma\|_T \leq \delta$ as required. This observation implies that the zero-finding method should be used to find an approximate solution of the perturbed trust-region subproblem $\min_{d \in \mathbb{R}^n} \{\, \mathcal{Q}(d) : \|d\|_T \leq \tilde{\delta} \,\}$. The solution of this problem is denoted by $\widetilde{d}$, i.e.,

$$\mathcal{Q}(\widetilde{d}) = \min_{d \in \mathbb{R}^n} \{\, \mathcal{Q}(d) : \|d\|_T \leq \tilde{\delta} \,\}, \tag{3.22}$$

and $\tilde{\sigma}$ is used to denote the nonnegative scalar associated with the optimality conditions

$$\big(H + \tilde{\sigma} T\big)\widetilde{d} = -g \ \text{ and } \ \tilde{\sigma}\big(\tilde{\delta} - \|\widetilde{d}\|_T\big) = 0 \tag{3.23}$$

of Result 3.1.1. The properties of an approximate solution of (3.22) are the focus of the next result.

Result 3.2.1 (Moré and Sorensen [25]). *Let $\tilde{\delta} = \delta/(1+\varepsilon)$ where ε is any scalar such that $0 < \varepsilon < 1$. Consider the vector s_σ such that*

$$(H + \sigma T)s_\sigma = -g \quad \textit{and} \quad \frac{|\,\|s_\sigma\|_T - \tilde{\delta}\,|}{\tilde{\delta}} \leq \varepsilon,$$

with $H+\sigma T$ positive semidefinite. Then s_σ satisfies the inequality

$$\mathscr{Q}(s_\sigma) \leq (1-\varepsilon)^2 \mathscr{Q}(\widetilde{d}), \quad \textit{where} \quad \mathscr{Q}(\widetilde{d}) = \min_{d\in\mathbb{R}^n} \left\{ \mathscr{Q}(d) : \|d\|_T \leq \tilde{\delta} \right\}. \tag{3.24}$$

Moreover, $\mathscr{Q}(s_\sigma)$ approximates the unique global minimum

$$\mathscr{Q}^* = \min_{d\in\mathbb{R}^n} \left\{ \mathscr{Q}(d) : \|d\|_T \leq \delta \right\}$$

in the sense that

$$\mathscr{Q}(s_\sigma) \leq \tau \mathscr{Q}^* \quad \textit{and} \quad \|s_\sigma\|_T \leq \delta, \tag{3.25}$$

with $\tau = (1-\varepsilon)^2/(1+\varepsilon)^2$.

Proof. The step s_σ solves the problem $\min_{d\in\mathbb{R}^n} \left\{ \mathscr{Q}(d) : \|d\|_T \leq \|s_\sigma\|_T \right\}$. From the definition of s_σ we have $\|s_\sigma\|_T \geq (1-\varepsilon)\tilde{\delta} \geq \|(1-\varepsilon)\widetilde{d}\|_T$, and hence

$$\mathscr{Q}(s_\sigma) \leq \mathscr{Q}\big((1-\varepsilon)\widetilde{d}\big) = (1-\varepsilon) g^{\mathrm{T}} \widetilde{d} + \tfrac{1}{2}(1-\varepsilon)^2 \widetilde{d}^{\mathrm{T}} H \widetilde{d}.$$

The optimality conditions (3.23) imply that $g^{\mathrm{T}}\widetilde{d} = -\widetilde{d}^{\mathrm{T}}(H+\tilde{\sigma}T)\widetilde{d} \leq 0$. Then, under the assumption that $(1-\varepsilon) < 1$, it must hold that $(1-\varepsilon)g^{\mathrm{T}}\widetilde{d} < (1-\varepsilon)^2 g^{\mathrm{T}}\widetilde{d}$ and

$$\mathscr{Q}(s_\sigma) \leq (1-\varepsilon)^2 \mathscr{Q}(\widetilde{d}) \quad \text{and} \quad \|s_\sigma\|_T \leq (1+\varepsilon)\tilde{\delta},$$

which establishes (3.24).

Let d^* be such that $\mathscr{Q}(d^*) = \mathscr{Q}^* = \min_{d\in\mathbb{R}^n} \left\{ \mathscr{Q}(d) : \|d\|_T \leq \delta \right\}$, where $\delta = (1+\varepsilon)\tilde{\delta}$. By definition, d^* must satisfy $\|d^*\|_T \leq (1+\varepsilon)\tilde{\delta}$, so that $\|d^*/(1+\varepsilon)\|_T \leq \tilde{\delta}$. The definition of $\widetilde{d}$ now yields the inequality

$$\mathscr{Q}(\widetilde{d}) \leq \mathscr{Q}\big(d^*/(1+\varepsilon)\big) = g^{\mathrm{T}} d^*/(1+\varepsilon) + \tfrac{1}{2} d^{*\mathrm{T}} H d^*/(1+\varepsilon)^2 \leq \mathscr{Q}(d^*)/(1+\varepsilon)^2.$$

Combining the bounds on $\mathscr{Q}(s_\sigma)$ and $\mathscr{Q}(\widetilde{d})$ gives

$$\mathscr{Q}(s_\sigma) \leq (1-\varepsilon)^2 \mathscr{Q}(\widetilde{d}) \leq \big((1-\varepsilon)/(1+\varepsilon)\big)^2 \mathscr{Q}^*,$$

and the result follows from the definition $\tau = (1-\varepsilon)^2/(1+\varepsilon)^2$. □

This result implies that the (implicit) condition for d to be an acceptable step becomes

$$\mathscr{Q}(d) \leq \tau \mathscr{Q}(d^*) \text{ and } \|d\|_T \leq (1+\varepsilon)\tilde{\delta}, \tag{3.26}$$

where $\tau = (1-\varepsilon)^2/(1+\varepsilon)^2$.

Next we consider the error induced by using an approximate null vector z in the degenerate case. Let σ_j be the current best estimate of $\tilde{\sigma}$, with $\sigma_j > 0$ and $\sigma_j > -\lambda_n$. Let s_j satisfy $(H+\sigma_j T)s_j = -g$. The resolution of the hard case discussed in Section 3.1 suggests a strategy in which the occurrence of an s_j such that $\|s_j\|_T < \tilde{\delta}$, initiates the search for a vector z_j for which the step $s_j + z_j$ satisfies $\|s_j + z_j\|_T = \tilde{\delta}$ and the decrease criterion (3.24). The complication is that λ_n is not known explicitly and it is not possible to define z_j so that $(H-\lambda_n T)z_j = 0$.

Let z_j be any vector such that $\|s_j + z_j\|_T = \tilde{\delta}$. We emphasize that, in general, z_j will not be a null vector of $H+\sigma_j T$. A suitable z_j can be viewed as a solution of a minimization problem. In particular, the identity $(H+\sigma_j T)s_j = -g$ and the definition of $\mathscr{Q}(d)$ may be combined to give the expression

$$\mathscr{Q}(s_j + z_j) = -\tfrac{1}{2}s_j^{\mathrm{T}}(H+\sigma_j T)s_j - \tfrac{1}{2}\sigma_j\tilde{\delta}^2 + \tfrac{1}{2}z_j^{\mathrm{T}}(H+\sigma_j T)z_j. \tag{3.27}$$

This implies that the z_j minimizing $\mathscr{Q}(d)$ is $z_j = \widetilde{d} - s_j$, i.e., choosing z_j so that $\widetilde{d} = s_j + z_j$ gives the exact minimum of the trust-region subproblem. However, as $\widetilde{d}$ is not known, we search for other values of z_j that make $\mathscr{Q}(d)$ small. One approach is to find a vector y_j such that $y_j^{\mathrm{T}}(H+\sigma_j T)y_j$ is a small as possible. The vector z_j is then taken to be τy_j where τ is a scalar such that $\|s_j + \tau y_j\|_T = \tilde{\delta}$. The expression (3.27) may be used to quantify how small $z_j^{\mathrm{T}}(H+\sigma_j T)z_j$ must be to provide an acceptable value of $s_j + z_j$. Because $z_j^{\mathrm{T}}(H+\sigma_j T)z_j \geq 0$ for all choices of z_j (including the choice $z_j = \widetilde{d} - s_j$) it must hold that

$$-\tfrac{1}{2}s_j^{\mathrm{T}}(H+\sigma_j T)s_j - \tfrac{1}{2}\sigma_j\tilde{\delta}^2 \leq \mathscr{Q}(\widetilde{d}).$$

Suppose that we can find a z_j such that

$$z_j^{\mathrm{T}}(H+\sigma_j T)z_j \le \varepsilon(2-\varepsilon)\big(s_j^{\mathrm{T}}(H+\sigma_j T)s_j + \sigma_j\tilde{\delta}^2\big). \tag{3.28}$$

Then this value may be used in equation (3.27) to obtain

$$\mathscr{Q}(s_j+z_j) \le -(1-\varepsilon)^2\big(\tfrac{1}{2}s_j^{\mathrm{T}}(H+\sigma_j T)s_j + \tfrac{1}{2}\sigma_j\tilde{\delta}^2\big) \le (1-\varepsilon)^2\mathscr{Q}(\widetilde{d}).$$

Therefore s_j+z_j satisfies the sufficient decrease condition (3.24) if z_j satisfies condition (3.28).

If λ_n is the least eigenvalue of $\widehat{H}$, then the vector u_n associated with the linear matrix pencil $Hu_n = \lambda_n N^{\mathrm{T}}Nu_n$ minimizes $z^{\mathrm{T}}(H+\sigma_j T)z$ and gives the corresponding minimum value $\lambda_n+\sigma_j$. If $H+\sigma_j T$ is nearly singular, i.e., if σ_j is close to $-\lambda_n$, then a vector z that makes $z^{\mathrm{T}}(H+\sigma_j T)z$ small can be found without the need to compute u_n (see Sections 3.2 and 3.5).

Properties of an approximate solution.

Let ε be a fixed scalar such that $0<\varepsilon<1$. Given the sequence $\{\sigma_j\}$, the scalar $\sigma=\sigma_j$ is considered to be an approximate zero of $\varphi(\sigma)$ (with $d=s_j$ the associated step) if the condition

($\mathbf{C}_1$) $|\widetilde{\psi}(\sigma_j)| \le \varepsilon$.

is satisfied.

In addition to this convergence condition, there are two situations in which the Newton iteration is terminated prior to convergence, in particular,

($\mathbf{T}_1$) if $\sigma_j=0$ and $\widetilde{\psi}(\sigma_j)<\varepsilon$; or

($\mathbf{T}_2$) if $\sigma_j>0$, $\widetilde{\psi}(\sigma_j)<-\varepsilon$ and there exists a sufficiently accurate approximate null vector z_j of $H+\sigma_j T$ (see the condition (3.30) below).

When termination occurs because of condition ($\mathbf{T}_1$), the scalar $\sigma=0$ and vector $d=s_j$ satisfy the optimality conditions of Result 3.1.1.

For termination under condition ($\mathbf{T}_2$), the vector s_j associated with the final σ_j is an approximation to $s^{\dagger}$ (see Result 3.1.8) and the approximate null vector z_j is scaled so that

$\|s_j + z_j\|_T = \delta$. In this case, we use σ_j and $s_j + z_j$ as approximate values of σ^* and d^*. Condition ($\mathbf{T}_2$) makes the algorithm well-defined when $\sigma^* = -\lambda_n$, and often allows the algorithm to terminate without the need to compute an accurate zero of $\varphi(\sigma)$. If $\sigma_j \in \big(-\lambda_n, \tilde{\sigma}\big]$, then the Newton iteration will converge to $\tilde{\sigma}$, but if $\tilde{\sigma}$ is close to $-\lambda_n$, then σ_j is likely to be greater than $\tilde{\sigma}$, and a safeguarded Newton method will require a large number of iterations to produce an iterate in $\big(-\lambda_n, \tilde{\sigma}\big]$. If $\sigma_j > \tilde{\sigma}$ then $\|s_j\|_T < \tilde{\delta}_j$ and if a z_j is computed in this situation, a vector $s_j + z_j$ is likely to be found that satisfies condition (3.28) in a few iterations, long before a safeguarded Newton method would produces an iterate in $\big(-\lambda_n, \tilde{\sigma}\big]$.

As long as $\sigma_j > \tilde{\sigma}$, the next iterate satisfies $\sigma_{j+1} < \sigma_j$. Thus if $\sigma_{j+1} > \tilde{\sigma}$, then the smallest eigenvalue of $H + \sigma_{j+1}T$ is positive and smaller than the smallest eigenvalue of $H + \sigma_j T$. The algorithm used to find a y such that $y^{\mathrm{T}}(H + \sigma_{j+1}T)y$ is small should be more likely to give a z_j such that $s_j + z_j$ is an acceptable step.

The algorithm also handles the case in which $\tilde{\sigma} = -\lambda_n$. In that case

$$\lim_{\sigma \to -\lambda_n^+} \psi(\sigma) < 0.$$

As $\sigma_j \geq -\lambda_n$, as long as $\sigma_j \neq -\lambda_n$, the vector s_j is defined and $\|Ns_j\| < \tilde{\delta}$. But then the safeguarded Newton method produces a sequence of matrices $H + \sigma_j T$ whose smallest eigenvalue remains positive but becomes increasingly close to zero.

Given δ and ε, the convergence and termination conditions given in the previous section provide a nonnegative scalar σ and vector $s_\sigma + z$ that satisfy the conditions

$$(H + \sigma T)s_\sigma = -g \quad \text{and} \quad \|s_\sigma + z\|_T \leq (1 + \varepsilon)\tilde{\delta}, \tag{3.29}$$

where $H + \sigma T$ is positive semidefinite, and the (possibly zero) vector z is chosen to satisfy the approximate null-vector condition

$$z^{\mathrm{T}}(H + \sigma T)z \leq \varepsilon(2 - \varepsilon)\big(s_\sigma^{\mathrm{T}}(H + \sigma T)s_\sigma + \sigma\tilde{\delta}^2\big). \tag{3.30}$$

More precisely, if we write $d = s_\sigma + z$, the cases may be summarized as follows. If $\sigma = 0$, then

$z = 0$ and the upper bound on $\|d\|_T$ is $(1-\varepsilon)\tilde{\delta}$; if $\sigma > 0$ and $\|d\|_T \geq (1-\varepsilon)\tilde{\delta}$ then $z = 0$ and $(1-\varepsilon)\tilde{\delta} \leq \|d\|_T \leq (1+\varepsilon)\tilde{\delta}$; finally, if $\sigma > 0$ and $\|d\|_T < (1-\varepsilon)\tilde{\delta}$, then $z \neq 0$ and $\|d\|_T = \tilde{\delta}$.

The next result summarizes the properties of the approximate solution and states that any d satisfying the conditions (3.29) and (3.30) also satisfies condition

$$\mathcal{Q}(d) \leq \tau \mathcal{Q}(d^*) \quad \text{and} \quad \|d\|_T \leq \delta, \tag{3.31}$$

for a certain τ and trust-region radius δ.

Lemma 3.2.1 (Moré and Sorensen [25]). *Let ε be a fixed scalar such that $0 < \varepsilon < 1$. Consider any $\sigma \geq 0$ and vector $s_\sigma + z$ satisfying* (3.29) *and* (3.30) *with $H + \sigma T$ positive semidefinite. Then $s_\sigma + z$ satisfies*

$$\mathcal{Q}(s_\sigma + z) \leq \tau \mathcal{Q}^* \ \text{ and } \ \|s_\sigma + z\|_T \leq \delta,$$

where $\tau = \big((1-\varepsilon)/(1+\varepsilon)\big)^2$, $\delta = (1+\varepsilon)\tilde{\delta}$ and $\mathcal{Q}^* = \min_{d \in \mathbb{R}^n} \big\{ \mathcal{Q}(d) : \|d\|_T \leq \delta \big\}$. ∎

This result implies that the approximate solution of the trust-region subproblem satisfies the decrease requirement (3.31) with a slightly larger value of the trust-region radius. Condition (3.31) is simpler to use when proving theoretical results, but conditions (3.29) and (3.30) are more appropriate for the discussion of the practical algorithm for the trust-region subproblem.

3.3 A safeguarded Newton iteration.

Next we consider a method for finding the vector $\widetilde{d}$ and scalar $\tilde{\sigma}$ of (3.23) based on finding a zero of the scalar-valued function

$$\widetilde{\varphi}(\sigma) = \frac{1}{\tilde{\delta}} - \frac{1}{\|s_\sigma\|_T}.$$

If a nonnegative $\widehat{\sigma}$ exists such that $\widetilde{\varphi}(\widehat{\sigma}) = 0$, then it may be computed efficiently using a safeguarded Newton method. In this scheme a sequence of intervals of uncertainty $I_j = [a_j, b_j]$ are constructed with $I_{j+1} \subset I_j$ and $\widehat{\sigma} \in I_j$. Clearly, if any b_j is negative, then it follows that

$\lambda_n > 0$ and the sequence can be terminated with $\widehat{\sigma} = 0$. If $\sigma_0 \in (-\lambda_n, \widehat{\sigma})$ then all subsequent iterates lie in $(-\lambda_n, \widehat{\sigma})$ and Newton's method converges.

In a conventional safeguarded iteration $\widetilde{\varphi}$ is not evaluated at a_j or b_j because it has already been computed there. In the trust-region calculation there is the additional consideration that a_j may define a "phantom root" associated with the root $\widehat{\sigma}$ such that $\widehat{\sigma} < -\lambda_n$. To allow for the hard case, in the initial stages when a_j remains fixed at zero and I_j is being reduced by decreasing b_j, we assume the possibility of a "phantom root" at $a_j = 0$.

The next result concerns the value of the function $\widetilde{\psi}(\sigma)$ used in the termination condition.

Lemma 3.3.1. *Suppose there is no $\widehat{\sigma} \in (-\lambda_n, \infty)$ such that $\widetilde{\psi}(\widehat{\sigma}) = 0$. If $\sigma_j > -\lambda_n$, then $\widetilde{\psi}(\sigma_j)$ is negative, and the Newton step $\sigma_{j+1}^N = \sigma_j - \widetilde{\psi}(\sigma_j)/\widetilde{\psi}(\sigma_j)$ satisfies $\sigma_{j+1}^N < -\lambda_n$. The same result holds for the Newton step $\sigma_{j+1}^N = \sigma_j - \widetilde{\varphi}(\sigma_j)/\widetilde{\varphi}(\sigma_j)$.* ∎

Corollary 3.3.1.1. *If $\widehat{\sigma} = 0$ and $\sigma_j > 0$, then $\sigma_{j+1}^N < 0$ and both $\widetilde{\varphi}(\sigma_j)$ and $\widetilde{\psi}(\sigma_j)$ are negative.* ∎

The goal of the safeguarding procedure is to provide a valid iterate in situations where $\sigma_{j+1}^N < -\lambda_n$ or $\sigma_{j+1}^N < 0$. The safeguarded iteration generates a nested sequence of half-open intervals $\{\mathscr{I}_j\}$ such that $\mathscr{I}_j = (a_j, b_j]$ with $\mathscr{I}_{j+1} \subset \mathscr{I}_j$. If σ_{j+1}^N is not a valid iterate, then σ_{j+1} is chosen as a positive weighted average of the endpoints a_{j+1} and b_{j+1} lying in the interior of $\mathscr{I}_{j+1}$.

Result 3.1.4 implies that if $\tilde{\sigma} \neq 0$ and $\tilde{\sigma} \neq -\lambda_n$, then $\tilde{\sigma} = \widehat{\sigma}$ and there is a nonempty interval $\mathscr{G} = (\max\{0, -\lambda_n\}, \widehat{\sigma}]$ of desirable starting points for which an unmodified Newton iteration will be well-defined and converge to $\widehat{\sigma}$. Suppose that $\mathscr{G}$ is nonempty. Then by design there is an interval $\widehat{\mathscr{G}} \subset \mathscr{G}$ of positive length such that $\widehat{\mathscr{G}} \subset \mathscr{I}_j$ for all j. In Theorem 3.3.3, it is shown that sequences $\{\mathscr{I}_j\}$ and $\{\sigma_j\}$ are generated so that if $\sigma_j \notin \mathscr{G}$, then $\sigma_j \in \mathscr{I}_j$ and $b_{j+2} - a_{j+2} \leq \gamma(b_j - a_j)$ for some $0 < \gamma < 1$. It follows that there is some finite iteration index q for which $\sigma_q \in \widehat{\mathscr{G}} \subset \mathscr{G}$. Care must be taken to ensure that the algorithm converges when $\mathscr{G}$ is

empty; i.e., when $\tilde{\sigma} = 0$ or $\tilde{\sigma} = -\lambda_n$. These cases are the subject of Theorems 3.3.4 and 3.3.5 respectively.

The algorithm requires routines $\bar{\lambda}_n(H)$ and $z_{\text{null}}(H, T, \sigma, s_j, \tilde{\delta})$. The routine $\bar{\lambda}_n(H)$ computes an estimate of the least eigenvalue of a symmetric matrix H, i.e., $\lambda_n(H) \leq \bar{\lambda}_n(H)$. The routine $z_{\text{null}}(H, T, \sigma, s_j, \tilde{\delta})$ computes a scaled approximate null-vector z_j of $H + \sigma T$. In particular, z_j is such that $z_j^{\mathrm{T}}(H + \sigma T)z_j$ is small and $\|s_j + z_j\|_T = \tilde{\delta}$. Then under suitable conditions, if $g \neq 0$ the following algorithm will produce a step d^* that satisfies condition (3.29).

We briefly describe the convergence properties of the safeguarding techniques.

Lemma 3.3.2. *Suppose g is nonzero. Define the interval $\mathscr{G} = \big(\max\{0, -\lambda_n\}, \tilde{\sigma}\big]$, where $\mathscr{G}$ is empty if $\tilde{\sigma} = 0$ or $\tilde{\sigma} = -\lambda_n$. Let q be the smallest iteration index for which $\sigma_q \in \mathscr{G}$, with $q = \infty$ if no such index exists. Suppose $\mathscr{G} \subset \mathscr{I}_0 = \big(a_0, b_0\big]$. Then for $i < q$, it holds that the intervals $\{\mathscr{I}_j\}$ are ordered by inclusion, that $\mathscr{G} \subset \mathscr{I}_j$ and that both $\sigma_j \in \mathscr{I}_j$ and $\tilde{\sigma} \in \mathscr{I}_j$. Moreover,*

$$b_{j+2} - a_{j+2} \leq \max\{\tfrac{1}{2}, 1 - \omega\}(b_j - a_j) \tag{3.32}$$

for all $i \geq 0$ for which $i + 2 < q$.

Proof. We proceed by induction. By assumption, $\mathscr{G} \subset \mathscr{I}_0$ and $\tilde{\sigma} \in \mathscr{I}_0$. The choice of σ_0 and $\mathscr{I}_0$ immediately gives $\sigma_0 \in \big(a_0, b_0\big]$.

Suppose $\mathscr{G} \subset \mathscr{I}_j$, $\sigma_j \in \mathscr{I}_j$ and $\tilde{\sigma} \in \mathscr{I}_j$. Three cases are possible. If $\sigma_j \in \mathscr{G}$, then $i \geq q$ and there is nothing to prove. Otherwise, it must hold that either $\sigma_j \leq -\lambda_n$ or $\sigma_j > \tilde{\sigma}$.

If $\sigma_j \leq -\lambda_n$, then σ_{j+1} is chosen by bisection and

$$b_{j+1} - a_{j+1} = \tfrac{1}{2}(b_j - \sigma_j) < \tfrac{1}{2}(b_j - a_j).$$

In this case, $\mathscr{I}_{j+1}$ is updated as $b_{j+1} = b_j$ and $a_{j+1} = \sigma_j \leq -\lambda_n$, and it follows that $\mathscr{G} \in \mathscr{I}_{j+1} \subset \mathscr{I}_j$, $\sigma_{j+1} \in \big(a_{j+1}, b_{j+1}\big]$ and $\tilde{\sigma} \in \big(a_{j+1}, b_{j+1}\big]$.

If $\sigma_j > \tilde{\sigma}$, then by Result 3.1.4, $\widetilde{\psi}(\sigma_j)$ is negative, with

$$[a_{j+1}, b_{j+1}] = [\max\{a_j, \bar{a}_j\}, \sigma_j], \tag{3.33}$$

Algorithm 3.2. Solution of the trust-region subproblem.

Specify constants $0 < \varepsilon < 1$, and $0 < \omega < 1$;
Choose $\sigma_0 \geq 0$;
$\bar{b} \leftarrow 1.05 \times \max\left(1, -\bar{\lambda}_n(\widehat{H})\right)/(1-\omega)$; $[a_0, b_0] \leftarrow [-1, \max\{\sigma_0, \bar{b}\}]$;
converged $\leftarrow$ false; $j \leftarrow 0$;
while not *converged* **do**
  Compute (n_+, n_-, n_0), the inertia of $H + \sigma_j T$;
  if $n_+ < n$ **then** [Check if $H + \sigma_j T$ is not positive definite]
    $[a_{j+1}, b_{j+1}] \leftarrow [\sigma_j, b_j]$; $\sigma_{j+1} \leftarrow \frac{1}{2}(a_{j+1} + b_{j+1})$;
  else [$H + \sigma_j T$ is positive definite]
    Solve $(H + \sigma_j T)s_j = -g$;
    if $|\widetilde{\psi}(\sigma_j)| < \varepsilon$ or $\left(\widetilde{\psi}(\sigma_j) < \varepsilon \text{ and } \sigma_j = 0\right)$ **then**
      $d \leftarrow s_j$; *converged* $\leftarrow$ true;
    else if $\widetilde{\psi}(\sigma_j) \leq -\varepsilon$ and $\sigma_j > 0$ **then**
      $z_j \leftarrow z_{\text{null}}\left(H, T, \sigma_j, s_j, \tilde{\delta}\right)$;
      if $z_j^{\mathrm{T}}(H + \sigma_j T)z_j > \varepsilon(2-\varepsilon)\left(s_j^{\mathrm{T}}(H + \sigma_j T)s_j + \sigma_j \tilde{\delta}^2\right)$ **then**
        $\bar{a}_j \leftarrow \sigma_j - z_j^{\mathrm{T}}(H + \sigma_j T)z_j / \|z_j\|_T^2$;
        $[a_{j+1}, b_{j+1}] \leftarrow \left[\max\{a_j, \bar{a}_j\}, \sigma_j\right]$;
      else
        $d \leftarrow s_j + z_j$; *converged* $\leftarrow$ true;
      end if
    else
      $[a_{j+1}, b_{j+1}] \leftarrow \left[a_j, b_j\right]$;
    end if
    if not *converged* **then**
      $\sigma_{j+1}^N \leftarrow \sigma_j - \widetilde{\varphi}(\sigma_j)/\widetilde{\varphi}'(\sigma_j)$;
      $\bar{\sigma}_{j+1} \leftarrow \max\{0, \sigma_{j+1}^N\}$;
      if $\bar{\sigma}_{j+1} > a_{j+1}$ **then**
        $\sigma_{j+1} \leftarrow \bar{\sigma}_{j+1}$
      else
        $\sigma_{j+1} \leftarrow \omega a_{j+1} + (1-\omega)b_{j+1}$;
      end if
    end if
  end if
  $j \leftarrow j + 1$;
end while

where $\bar{a}_j = \sigma_j - z^{\mathrm{T}}(H+\sigma_j I)z/\|z\|_T^2$. It is not difficult to see that $\bar{a}_j \leq -\lambda_n \leq 0$ (see Moré and Sorensen [25] for details). Hence $\mathscr{G} \subset \mathscr{I}_{j+1} \subset \mathscr{I}_j$. Moreover, $\sigma_{j+1}^N < \tilde{\sigma} < \sigma_j \leq b_j$ and so $\bar{\sigma}_{j+1} \leq \sigma_j$. However, in this case the rule

$$\sigma_{j+1} = \begin{cases} \bar{\sigma}_{j+1} & \text{if } \bar{\sigma}_{j+1} > a_j; \\ \omega a_{j+1} + (1-\omega) b_{j+1} & \text{otherwise,} \end{cases}$$

implies that $\sigma_{j+1} \in \left(a_{j+1}, b_{j+1}\right]$. Hence, for $i < q$ we may conclude that the intervals $\left\{\mathscr{I}_j\right\}$ are ordered by inclusion, that $\mathscr{G} \subset \mathscr{I}_j$ and that both $\sigma_j \in \mathscr{I}_j$ and $\tilde{\sigma} \in \mathscr{I}_j$. It remains to show that the inequality (3.32) holds.

Now consider the length of $\mathscr{I}_{j+2}$. Observe that for any $\ell > 0$, if $\sigma_\ell \leq \tilde{\sigma}$ then either $\sigma_\ell \in \mathscr{G}$ or $\sigma_\ell < -\lambda_n$. If $\sigma_\ell \in \mathscr{G}$ then all subsequent iterates are in $\mathscr{G}$ and the inequality (3.32) is irrelevant. If $\sigma_\ell < -\lambda_n$ then both $\sigma_{\ell+1}$ and $\left(a_{\ell+1}, b_{\ell+1}\right]$ are chosen by bisection and the inequality (3.32) will hold for both iterations $i = \ell$ and $i = \ell - 1$.

Thus, we need only consider the case in which both $\sigma_j > \tilde{\sigma}$ and $\sigma_{j+1} > \tilde{\sigma}$. If $\sigma_j > \tilde{\sigma}$, then two situations are possible. If $\sigma_{j+1} = \max\left\{0, \sigma_{j+1}^N\right\}$, then from Result 3.1.4 and the assumption that $\tilde{\sigma} > 0$, it follows that $\sigma_{j+1} \leq \tilde{\sigma}$. Otherwise, σ_{j+1} is defined by the rule $\sigma_{j+1} = \omega a_{j+1} + (1-\omega) b_{j+1}$. Suppose $\sigma_{j+1} > \tilde{\sigma}$. In this case, $\widetilde{\psi}(\sigma_{j+1})$ is negative, the interval $\mathscr{I}_{j+2}$ is defined by (3.33) and

$$b_{j+2} - a_{j+2} > \sigma_{j+1} - a_{j+1} = (1-\omega) b_{j+1} + \omega a_{j+1} - a_{j+1} \geq (1-\omega)(b_j - a_j).$$

Thus (3.32) holds and the lemma is proved. □

Theorem 3.3.3. *Suppose g is nonzero. Define the interval $\mathscr{G} = \left(\max\left\{0, -\lambda_n\right\}, \tilde{\sigma}\right]$. If $\max\left\{0, -\lambda_n\right\} < \tilde{\sigma}$ then Algorithm* 3.2 *will produce an iterate $\sigma_q \in \mathscr{G}$ or terminate under conditions* ($\mathbf{T}_1$) *or* ($\mathbf{T}_2$) *before that occurs.*

Proof. Assume that Algorithm 3.2 does not terminate before producing an iterate $\sigma_q \in \mathscr{G}$.

If $b_0 \geq \tilde{\sigma}$, then the conditions of Lemma 3.3.2 are met. But then, because $\tilde{\sigma} >$

$\max\{0, -\lambda_n\}$, the interval $\mathscr{G}$ has a positive length and so the bound (3.32) together with the inclusions $\sigma_0 \in (a_0, b_0]$ and $\mathscr{G} \subset \mathscr{I}_0$ imply that q is finite.

If, on the other hand, $b_0 < \tilde{\sigma}$, then either $\sigma_0 \in \mathscr{G}$ or $\sigma_0 \leq -\lambda_n$. In the latter case, the iterates are chosen by repeatedly bisecting the intervals $\mathscr{I}_j$ until $\sigma_j > -\lambda_n$, and hence $\sigma_j \in \mathscr{G}$. □

If $\tilde{\sigma} = 0$ or $\tilde{\sigma} = -\lambda_n$, then $\mathscr{G}$ is empty and Theorem 3.3.3 does not apply. The case in which $\tilde{\sigma} = 0$ is a desirable special case, because then d is an unmodified Newton iterate for the underlying optimization algorithm. Therefore, Algorithm 3.2 has been designed to favor $\sigma = 0$ as a solution to the subproblem.

Theorem 3.3.4. *If $\tilde{\sigma} = 0$, then either σ_0 or σ_1 is zero.*

Proof. If $\sigma_0 \neq 0$ and $\tilde{\sigma} = 0$, then $\sigma_0 > \tilde{\sigma} \geq -\lambda_n$ and so $\bar{\sigma}_1 = \max\{0, \sigma_1^N\}$ is defined. But by Corollary 3.3.1.1, the Newton step σ_1^N is negative and hence $\bar{\sigma}_1 = 0$. Furthermore, $a_1 = \max\{a_1, \bar{a}_1\}$, where $\bar{a}_1 = \sigma_0 - z_1^{\mathrm{T}}(H + \sigma_1 T)z_1/\|z_1\|_T^2$. As discussed in the proof of Theorem 3.3.3, $\bar{a}_1 \leq -\lambda_n$. Therefore $a_{j+1} \leq 0 = \bar{\sigma}_1$, and hence $\sigma_1 = 0$. □

If $\tilde{\sigma} = 0 > -\lambda_n$, then $\sigma = 0$ and $\widetilde{d} = s_\sigma = -H^{-1}g$ satisfy conditions (3.29) and (3.30), and Algorithm 3.2 will terminate. Neither Theorem 3.3.3 nor Theorem 3.3.4 implies convergence when $\tilde{\sigma} = -\lambda_n$. Theorem 3.3.3 cannot be applied because $\mathscr{G}$ is empty. Theorem 3.3.4 implies that if $\tilde{\sigma} = -\lambda_n = 0$, then either σ_0 or σ_1 will be zero. However, if $\lambda_n = 0$ then H is not invertible and s_σ cannot be defined as $-H^{-1}g$.

Theorem 3.3.5. *If $\tilde{\sigma} = -\lambda_n$ then either* Algorithm 3.2 *finds a point satisfying the convergence criteria in a finite number of iterations, or $\sup_{i \geq j} \sigma_j$ converges to $\tilde{\sigma}$ from above.*

Proof. Suppose the algorithm does not find a point that satisfies the convergence criteria. The interval $\mathscr{G}$ is empty and so the conditions of Lemma 3.3.2 are trivially met. We may therefore conclude that $\lim_{j\to\infty} \sigma_j = \tilde{\sigma}$. We must now show that $\sigma_j > \tilde{\sigma}$ infinitely often. But if $\sigma_j \leq \tilde{\sigma} = -\lambda_n$, subsequent iterates are chosen by bisection until one is greater than $-\lambda_n$. □

It is important that $\sigma_j > \tilde{\sigma}$ holds infinitely often, because $z_{\text{null}}(\cdot)$ is only defined for $\sigma_j > -\lambda_n$. If the $z_{\text{null}}(\cdot)$ routine can be guaranteed to produces a z_j satisfying

$$z_j^{\mathrm{T}}(H+\sigma_j T)z_j \leq \varepsilon(2-\varepsilon)\left(s_j^{\mathrm{T}}(H+\sigma_j T)s_j + \sigma_j\tilde{\delta}^2\right)$$

as σ_j converges to $-\lambda_n$ from above, then Theorem 3.3.5 guarantees that Algorithm 3.2 must terminate after finitely many iterations.

3.4 The Implementation of the Moré-Sorensen Method

The method of Moré and Sorensen uses the Cholesky factorization $R_j^{\mathrm{T}}R_j = H + \sigma_j T$ to compute the vectors s_j and z_j. The vector y_j that makes $y_j^{\mathrm{T}}(H+\sigma_j T)y_j$ as small as possible is the eigenvector corresponding to the smallest eigenvalue of $H+\sigma_j T$ (see (3.3)). The corresponding minimum value of $y_j^{\mathrm{T}}(H+\sigma_j T)y_j$ is the smallest eigenvalue of $H+\sigma_j T$. However, it is not necessary to solve an eigenproblem to find an acceptable y_j. Moré and Sorensen generate an appropriate y_j by employing a method proposed by Cline, Moler, Stewart and Wilkinson [4] to estimate the condition number of a matrix. An implementation of this method is included in the software library LINPACK.

Given a matrix A and its Cholesky factorization $R^{\mathrm{T}}R$, the LINPACK algorithm attempts to find a vector b of unit norm and a vector $w = A^{-1}b$ for which $\|w\|$ is as large as possible. The vector y is then w normalized. The vector b is chosen to be a vector with entries equal to ± 1, where the sign of each element of b is chosen to cause growth during the process of solving the system $R^{\mathrm{T}}v = b$. The vector w is then found by solving the system $Rw = v$. The method is somewhat more sophisticated than simply choosing b_i to maximize $\widehat{u}_i$ because many simple examples were found that caused the obvious method to fail. Furthermore, the authors frequently rescale v and b to avoid overflow. A forward and backward substitution each requires $O(\frac{1}{2}n^2)$ multiplications, and so the entire process of finding an appropriate z requires work proportional to n^2 and may be considered relatively inexpensive.

Moré and Sorensen prove the value of y found by the algorithm of Cline et. al. will eventually produce a z_j that satisfies (3.28) and consequently and $s_j + z_j$ that satisfies condition (3.26). In this way, the case in which $\tilde{\sigma} = -\lambda_n$ mimics the usual behavior of the algorithm when $\tilde{\sigma} \approx -\lambda_n$.

We observe that although in some cases $\tilde{\sigma} = -\lambda_n$, we reject out of hand any proposed iterate σ_+ for which $\sigma_+ = -\lambda_n$. The safeguarding algorithm then produces the next value of σ_+, and the procedure outlined above eventually finds an acceptable z_j and an acceptable s_j. While in theory if $\sigma_+ = -\lambda_n$ and $(H + \sigma_+ T)s = -g$ is compatible, we might attempt to find an acceptable value of s, in practice, it is usually not possible to determine numerically whether a given system $(H + \sigma_+ T)s = -g$ is exactly singular but compatible. A variation on Moré and Sorensen's algorithm might attempt to search for an acceptable s in the rare instance in which $\sigma_+ = -\lambda_n$ and $(H + \sigma_+ T)s = -g$ is determined numerically to be compatible. Such a modification to the algorithm seems unlikely to have any positive practical effect.

3.5 Adapting Trust-Region Methods for Constrained Problems

The proposed method involves two modifications to the Moré and Sorensen's algorithm. The first modification is fundamental for the efficient solution of the constrained problem. The conventional Moré-Sorensen method cannot be used in the constrained case because the trust-region matrix H is the approximate Hessian of the primal-dual merit function, which has a doubly-augmented structure. This makes it impractical to compute a Cholesky factorization of $H + \sigma T$. Instead, systems of the form $(H + \sigma T)s = -g$ are solved by factoring a transformed system. As an alternative, we propose that an approximate null vector be determined as a by-product of a method proposed by Hager [21] (and modified by Higham [23]) to estimate the one-norm condition number of an $n \times n$ matrix B. Given an $n \times n$ matrix B, Hager's method

computes an estimate γ of the norm

$$\|B\|_1 = \max_{x \neq 0} \frac{\|Bx\|_1}{\|x\|_1} = \max_{\|x\|_1 = 1} \|Bx\|_1.$$

Hager's algorithm attempts to compute $\|B\|_1$ efficiently without computing the elements of B. In particular, only matrix-vector products of the form Bv and $B^{\mathrm{T}}v$ are required.

A by-product of the computation is an estimate $y = Bw$, where $\gamma = \|y\|_1 / \|w\|_1 = \|y\|_1 / \|B^{-1}y\|_1$. If $B = A^{-1}$ and γ is large, then

$$\frac{\|Ay\|_1}{\|y\|_1} = \frac{1}{\gamma},$$

and y is an approximate null vector of A. Hager's method requires at most five matrix-vector products, which makes the work proportional to a small multiple of n^2.

Hager's method

For any $n \times n$ matrix B, the one-norm $\|B\|_1$ is the global maximum of the convex function

$$f(x) = \|Bx\|_1 \text{ over the convex set } \Omega = \left\{ x : \|x\|_1 \leq 1 \right\}.$$

This suggests using an optimization approach to estimate $\|B\|_1$, i.e., iteratively move from one point in Ω to another where f is greater, testing for optimality at each stage. Hager [21] derives such an algorithm by exploiting properties of f and Ω.

Algorithm 3.3. Hager's method for maximizing $\|A^{-1}x\|_1 / \|x\|_1$.

Choose x with $\|x\|_1 = 1$;
repeat
  Solve $Ay = x$; $\gamma = \|y\|_1$; [y is $A^{-1}x$]
  $\xi = \mathrm{sign}(y)$;
  Solve $A^{\mathrm{T}}g = \xi$; [g is a subgradient of $\|A^{-1}x\|_1$]
  $x = e_j$, where $g_j = \|y\|_\infty$;
until $\|g\|_\infty \leq g^{\mathrm{T}}x$

Algorithm 3.4 gives the Hager-Higham method for computing an approximate null-vector of a matrix A. Algorithm 3.4 computes at most five such products, and so the work associated

Algorithm 3.4. The Higham-Hager method for maximizing $\|A^{-1}x\|_1/\|x\|_1$.

Solve $Ay = \frac{1}{n}e$; $\gamma = \|y\|_1$;
if $n = 1$ **then** stop;
$\xi = \text{sign}(y)$; Solve $A^{\mathrm{T}}g = \xi$; [g is a subgradient of $\|A^{-1}y\|_1$]
$k = 2$;
repeat
  $j = \min\{\, i : |g_i| = \|g\|_\infty \,\}$;
  Solve $Ay = e_j$;
  $\bar{\gamma} = \gamma$; $\gamma = \|y\|_1$;
  if $\text{sign}(y) = \xi$ or $\gamma \le \bar{\gamma}$ **then** break; [avoids repeating the solve]
  $\xi = \text{sign}(y)$; Solve $A^{\mathrm{T}}g = \xi$; [g is a subgradient of $\|A^{-1}y\|_1$]
  $k \leftarrow k+1$;
until $\|g\|_\infty = g_j$ or $k > 5$

for $i \leftarrow 1 : n$ **do**
  $x_i = (-1)^{i+1}\left(1 + \dfrac{i-1}{n-1}\right)$;
end for;
Solve $Au = x$; [$\|x\|_1 = \frac{3}{2}n$]
if $2\|u\|_1/(3n) > \gamma$ **then**
  $y = u$; $\gamma = 2\|u\|_1/(3n)$;
end if

with Hager's algorithm proportional to a small multiple of n^2. If $N \neq I$, the Algorithm 3.4 will often compute better values of y than the LINPACK algorithm. The LINPACK algorithm uses a Cholesky factorization of $H + \sigma T$ to attempt to find a vector y such that $\|y\| = 1$ and $y^{\mathrm{T}}(H + \sigma T)y$ is a small as possible. A lower bound on λ_n is then

$$\bar{\lambda} = \frac{y^{\mathrm{T}}(H + \sigma T)y}{\|Ny\|^2} - \sigma.$$

From this expression, however, it is clear that $\bar{\lambda}$ will usually be a better bound on $-\lambda_n$ if y is chosen so that $\|Ny\| = 1$ and $y^{\mathrm{T}}(H + \sigma T)y$ is as small as possible. The LINPACK algorithm is not very effective at computing such a y from a Cholesky factorization of $H + \sigma T$. On the other hand, given a factorization of $H + \sigma T$ we may easily compute such a y by performing Hager's algorithm on the system $N^{-\mathrm{T}}(H + \sigma T)N^{-1}$ and transforming the result. It is clear how to compute matrix-vector products with $N(H + \sigma T)^{-1}N$ given a Cholesky factorization of $H + \sigma T$.

The disadvantage of using Hager's algorithm is theoretical. The algorithm for computing a solution for the trust-region subproblem has been proven to converge for any value of $\tilde{\sigma}$ if the algorithm of Cline et. al. is used to find a value of z_j. We are aware of no comparable proof for Hager's algorithm. The difficulty is that Hager's algorithm may be fooled, and terminate with a w for which $\|w\|_1 \neq \|A^{-1}\|_1$. Higham's modifications make the algorithm more likely to succeed, and make it likely that $\|w\|_1$ is large even if $\|w\|_1 \neq \|A^{-1}\|$. The algorithm of Cline et al. may also be fooled into producing poor values of y. Still, Moré and Sorensen are able to show that if $\tilde{\sigma} = -\lambda_n$, and the LINPACK algorithm is used to find z_j, then $z_j^{\mathrm{T}}(H+\sigma_j T)z_j \to 0$. We have observed good performance from Hager's algorithm. Moreover, while Moré and Sorensen do show that with the LINPACK algorithm, $z_j^{\mathrm{T}}(H+\sigma_j T)z_j \to 0$, their proof presents a worst case rate of convergence that is so slow that the algorithm would be completely impractical if it ever approached that rate of convergence. Thus, while we would prefer to have a theoretical guarantee of convergence, practical considerations lead us to prefer Hager's algorithm.

The second modification is minor. Whenever $\|s_j\|_T < \tilde{\delta}$, the procedure outlined above is used to attempt to find a z_j and an acceptable step $d_j = s_j + z_j$. However, if $\|s_j\|_T \geq (1-\varepsilon)\tilde{\delta}$, then $d_j = s_j$ is already an acceptable step. In this situation, we take $d_j = s_j$. Moré and Sorensen compute z_j and compare $\mathscr{Q}(s_j)$ to $\mathscr{Q}(s_j + z_j)$. They then let d_j be the vector that produces the smaller value of $\mathscr{Q}$. We choose not to compute z_j because this choice makes the logic of the algorithm simpler, and because s_j is the minimizer of $\mathscr{Q}(d)$ subject to $\|d\|_T \leq \|s\|_T$. Theory does not provide any compelling reason for making either choice.

Chapter 4

Primal-Dual Methods for Constrained Problems with Slacks

In this chapter, we consider the use of primal-dual shifted penalty-barrier methods for the solution of optimization problems with nonlinear inequality constraints. Although the focus is on the treatment of inequality constraints, it must be emphasized that the methods are easily extended to problems with a mixture of equality and inequality constraints. A merit function is defined as the sum of the objective function and certain shifted penalty and barrier functions. A fundamental property of the merit function is that an unconstrained minimizer satisfies perturbed KKT conditions for the constrained problem with perturbations equal to the values of the penalty and barrier parameters.

The problem to be solved is

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

where $c : \mathbb{R}^n \to \mathbb{R}^m$ and $f : \mathbb{R}^n \to \mathbb{R}$ are both smooth. In order to avoid the need to find a strictly feasible point for the constraints of (NIP), each inequality $c_i(x) \geq 0$ is written in terms of an equality and nonnegative slack variable $c_i(x) - s_i = 0$ and $s_i \geq 0$. This gives the equivalent problem

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

The vector (x^*, s^*, y^*, w^*) is called a first-order KKT point for problem (NIPs) when

$$c(x^*) - s^* = 0, \qquad s^* \geq 0, \tag{4.1a}$$

$$g(x^*) - J(x^*)^{\mathrm{T}} y^* = 0, \qquad y^* - w^* = 0, \tag{4.1b}$$

$$s^* \cdot w^* = 0, \qquad w^* \geq 0. \tag{4.1c}$$

The vectors y^* and w^* constitute the Lagrange multiplier vectors for, respectively, the equality constraints $c(x) - s = 0$ and non-negativity constraints $s \geq 0$. The vector (x_k, s_k, y_k, w_k) will be used to denote the k-th primal-dual iterate computed by the proposed algorithm, with the aim of giving limit points of $\left\{ (x_k, s_k, y_k, w_k) \right\}_{k=0}^{\infty}$ that are first-order KKT points for problem (NIPs), i.e., limit points that satisfy (4.1).

An important concept related to the design of efficient algorithms for computing first-order KKT points for problem (NIPs) is that of perturbed optimality conditions. An appropriate set of perturbed conditions for (4.1) is given by

$$\begin{aligned} g(x) - J(x)^{\mathrm{T}} y = 0, \qquad & y - w = 0, \\ c(x) - s = \mu^P (y^E - y), \qquad & s \geq 0, \\ s \cdot w = \mu^B (w^E - w), \qquad & w \geq 0, \end{aligned} \tag{4.2}$$

where $y^E \in \mathbb{R}^m$ is an estimate of a Lagrange multiplier vector for the constraint $c(x) - s = 0$, $w^E \in \mathbb{R}^m$ is an estimate of a Lagrange multiplier for the constraint $s \geq 0$, and the scalars μ^P and μ^B are positive penalty and barrier parameters, respectively. (The interpretation of μ^P and μ^B as penalty and barrier parameters is discussed below.) In the neighborhood of a first-order KKT point it is well-known that computing the search direction as the solution of the Newton equations for a zero of the perturbed optimality conditions provides the favorable local convergence rate associated with Newton's method. At the same time, to ensure convergence to a first-order KKT point from an arbitrary starting point, an algorithm must include a strategy for deciding when one iterate is preferable to another. These considerations motivate the formulation of the new

shifted primal-dual penalty-barrier function

$$M(x,s,y,w;y^E,w^E,\mu^P,\mu^B) = \underbrace{f(x) - (c(x)-s)^{\mathrm{T}}y^E}_{(A)\qquad(B)}$$
$$\underbrace{+\frac{1}{2\mu^P}\|c(x)-s\|^2}_{(C)} \underbrace{+\frac{1}{2\mu^P}\|c(x)-s+\mu^P(y-y^E)\|^2}_{(D)}$$
$$\underbrace{-\sum_{i=1}^{m}\mu^B w_i^E \ln\left(s_i+\mu^B\right)}_{(E)} \underbrace{-\sum_{i=1}^{m}\mu^B w_i^E \ln\left(w_i(s_i+\mu^B)\right)}_{(F)} \underbrace{+\sum_{i=1}^{m} w_i(s_i+\mu^B)}_{(G)}.$$

It is shown in Section 4.1.3 that in the neighborhood of a minimizer of (NIPs) satisfying certain second-order optimality conditions, the Newton equations for a zero of the perturbed optimality conditions (4.2) are equivalent to the Newton equations for a minimizer of M. Also, it is shown in Section 4.1 that if the parameters y^E, w^E, μ^P, and μ^B are updated appropriately, then stationary points of M have properties that may be used in the formulation of a globally convergent algorithm for (NIPs).

Let S and W denote diagonal matrices with diagonal entries s and w (i.e., $S = \operatorname{diag}(s)$ and $W = \operatorname{diag}(w)$) such that $s_i + \mu^B > 0$ and $w_i > 0$. Define the positive-definite matrices

$$D_P = \mu^P I \quad \text{and} \quad D_B = (S+\mu^B I)W^{-1},$$

and auxiliary vectors

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

Then $\nabla M(x,s,y,w;y^E,w^E,\mu^P,\mu^B)$ may be written as

$$\nabla M = \begin{pmatrix} g - J^{\mathrm{T}}\big(\pi^Y + (\pi^Y - y)\big) \\ \big(\pi^Y - y\big) + \big(\pi^Y - \pi^W\big) + \big(w - \pi^W\big)\big) \\ -D_P(\pi^Y - y) \\ -D_B(\pi^W - w) \end{pmatrix}, \tag{4.3}$$

with $g = g(x)$ and $J = J(x)$. The purpose of writing the gradient ∇M in this form is to highlight the quantities $\pi^Y - y$ and $\pi^W - w$, which are important in the analysis. Similarly, the penalty-barrier function Hessian $\nabla^2 M(x,s,y,w;y^E,w^E,\mu^P,\mu^B)$ is written in the form

$$\nabla^2 M = \begin{pmatrix} H + 2J^{\mathrm{T}} D_P^{-1} J & -2J^{\mathrm{T}} D_P^{-1} & J^{\mathrm{T}} & 0 \\ -2D_P^{-1} J & 2(D_P^{-1} + D_B^{-1} W^{-1} \Pi^W) & -I & I \\ J & -I & D_P & 0 \\ 0 & I & 0 & D_B W^{-1} \Pi^W \end{pmatrix}, \tag{4.4}$$

where $H = H\big(x, \pi^Y + (\pi^Y - y)\big)$ and $\Pi^W = \operatorname{diag}(\pi^W)$.

In developing algorithms, the goal is to achieve rapid convergence to a solution of (NIPs) without the need for μ^P and μ^B to go to zero. The underlying mechanism for ensuring convergence is the minimization of M for fixed parameters.

4.1 A Modified Newton Method

This section concerns the properties of a modified Newton method for the minimization of M for fixed parameters y^E, w^E, μ^P and μ^B. In this case the notation can be simplified by omitting the reference to y^E, w^E, μ^P and μ^B when writing M, ∇M and $\nabla^2 M$.

At the start of iteration k, given the primal-dual iterate $v_k = (x_k, s_k, y_k, w_k)$, the search direction $d_k = (\Delta x_k, \Delta s_k, \Delta y_k, \Delta w_k)$ is computed by solving the linear system of equations

$$\widehat{H}_k^M d_k = -\nabla M(v_k), \tag{4.5}$$

where $\widehat{H}_k^M$ is a positive-definite approximation of the matrix $\nabla^2 M(x_k, s_k, y_k, w_k)$. (The definition of $\widehat{H}_k^M$ and the properties of the equations (4.5) are discussed in Section 4.1.1 below.)

Subsection 4.1.1 focuses on the properties of the modified-Newton matrix, while subsection 4.1.2 discusses an efficient method for solving the resulting modified-Newton equations for the primal-dual search direction. Finally, subsection 4.1.3 establishes the relationship between the computed search direction and a shifted variant of the conventional primal-dual

path-following equations. As this section is concerned with details of only a single iteration, the notation is simplified by omitting the dependence on the iteration k. In particular, we write $v = v_k$, $y^E = y_k^E$, $w^E = w_k^E$, $\pi^Y = \pi_k^Y$, $\pi^W = \pi_k^W$, $d = d_k$, $c = c(x_k)$, $J = J(x_k)$, $g = g(x_k)$, $D_P = \mu_k^P I$, $D_B = (S_k + \mu_k^B I)W_k^{-1}$, and $\widehat{H}^M = \widehat{H}_k^M$.

4.1.1 Definition of the modified-Newton matrix

The choice of $\widehat{H}^M$ in the equations $\widehat{H}^M d = -\nabla M(v)$ is based on making two modifications to $\nabla^2 M$. The first involves substituting y for π^Y and w for π^W in (4.4). (Lemma 4.2.2 and the discussion of subsection 4.1.3 below provide justification for this choice.) The second modification is to replace the modified Hessian $H(x,y)$ by a symmetric $\widehat{H}$ such that $\widehat{H} \approx H(x,y)$ and $\widehat{H}^M$ is positive definite. These modifications give an $\widehat{H}^M$ in the form

$$\widehat{H}^M = \begin{pmatrix} \widehat{H} + 2J^{\mathrm{T}} D_P^{-1} J & -2J^{\mathrm{T}} D_P^{-1} & J^{\mathrm{T}} & 0 \\ -2D_P^{-1} J & 2(D_P^{-1} + D_B^{-1}) & -I & I \\ J & -I & D_P & 0 \\ 0 & I & 0 & D_B \end{pmatrix}. \tag{4.6}$$

Practical conditions for the choice of a positive-definite $\widehat{H}$ are based on the next result, which is established in Gill, Kungurtsev & Robinson [13].

Theorem 4.1.1. *The matrix $\widehat{H}^M$ in* (4.6) *is positive definite if and only if*

$$\mathrm{In}(K) = \mathrm{In}(n,m,0), \;\; \textit{where} \;\; K = \begin{pmatrix} \widehat{H} & J^{\mathrm{T}} \\ J & -(D_B + D_P) \end{pmatrix}, \tag{4.7}$$

which holds if and only if $\widehat{H} + J^{\mathrm{T}}(D_P + D_B)^{-1} J^{\mathrm{T}}$ is positive definite. □

There are a number of alternative approaches for choosing $\widehat{H}$ based on computing a factorization of the $(n+m)$ by $(n+m)$ matrix K (4.7) (see, e.g., Gill and Robinson [15, Section 4], Forsgren [9], Forsgren and Gill [10], Gould [19], Gill and Wong [16], and Wächter and Biegler [31]). All of these methods use $\widehat{H} = H(x,y)$ if this gives a sufficiently positive-definite $\widehat{H}^M$. The next result

shows that $\widehat{H} = H(x,y)$ gives a positive-definite $\widehat{H}^M$ in a sufficiently small neighborhood of a solution satisfying second-order sufficient optimality conditions and strict complementarity. The proof can be found in Gill, Kungurtsev & Robinson [13].

Theorem 4.1.2. *The matrix $\widehat{H}^M$ in* (4.6) *with the choice $\widehat{H} = H(x,y)$ is positive definite for all $u = (x,s,y,w,y^E,w^E,\mu^P,\mu^B)$ sufficiently close to $u^* = (x^*, s^*, y^*, w^*, y^*, w^*, 0, 0)$, when (x^*, s^*, y^*, w^*) is a solution of problem* (NIPs) *that satisfies second-order sufficient optimality conditions and strict complementarity.* □

4.1.2 Solving the modified-Newton equations

The modified-Newton equations (4.5) defined with $\widehat{H}^M$ from (4.6) should *not* be solved directly because of the potential for numerical instability. Instead, an *equivalent* transformed system should be solved based on the transformation

$$U = \begin{pmatrix} I & 0 & -2J^{\mathrm{T}}D_P^{-1} & 0 \\ 0 & I & 2D_P^{-1} & -2D_B^{-1} \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & W \end{pmatrix}.$$

As U is nonsingular, the modified-Newton direction d from (4.5) satisfies

$$U\widehat{H}^M d = -U\nabla M(x,s,y,w;y^E,w^E,\mu^P,\mu^B),$$

which, upon multiplication and application of the identity $WD_B = S + \mu^B I$, yields

$$\begin{pmatrix} \widehat{H} & 0 & -J^{\mathrm{T}} & 0 \\ 0 & 0 & I & -I \\ J & -I & D_P & 0 \\ 0 & W & 0 & S+\mu^B I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} g - J^{\mathrm{T}}y \\ y - w \\ c - s + \mu^P(y - y^E) \\ s \cdot w + \mu^B(w - w^E) \end{pmatrix}. \tag{4.8}$$

The solution of this transformed system may be found by solving two sets of equations, one diagonal and the other of order $n+m$. To see this, first observe that the equations (4.8) may be

written in the form

$$\begin{pmatrix} \widehat{H} & 0 & -J^{\mathrm{T}} & 0 \\ 0 & 0 & I & -I \\ J & -I & D_P & 0 \\ 0 & I & 0 & D_B \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} g - J^{\mathrm{T}} y \\ y - w \\ c - s + \mu^P (y - y^E) \\ W^{-1}\big(s \cdot w + \mu^B (w - w^E)\big) \end{pmatrix}. \tag{4.9}$$

The solution of (4.9) is given by

$$\Delta w = y - w + \Delta y \quad \text{and} \quad \Delta s = -W^{-1}\big(s \cdot (y + \Delta y) + \mu^B (y + \Delta y - w^E)\big), \tag{4.10}$$

where Δx and Δy satisfy the equations

$$\begin{pmatrix} \widehat{H} & -J^{\mathrm{T}} \\ J & D_P + D_B \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} g - J^{\mathrm{T}} y \\ c - s + \mu^P (y - y^E) + W^{-1}\big(s \cdot y + \mu^B (y - w^E)\big) \end{pmatrix},$$

or, equivalently, the symmetric equations

$$\begin{pmatrix} \widehat{H} & J^{\mathrm{T}} \\ J & -(D_P + D_B) \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} g - J^{\mathrm{T}} y \\ D_P (y - \pi^Y) + D_B \big(y - \pi^W\big) \end{pmatrix}. \tag{4.11}$$

Solving this $(n+m) \times (n+m)$ symmetric system is the dominant cost of an iteration. The identity $w + \Delta w = y + \Delta y$ implies that if the initial values satisfy $y_0 = w_0$ and $y_0^E = w_0^E$, and the positive safeguarding values in (4.15) satisfy $y_{\max} = w_{\max}$, then all subsequent iterates will satisfy $w = y$.

4.1.3 Relationship to primal-dual path-following

Consider the perturbed optimality conditions (4.2) and their associated primal-dual path-following equations

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

A zero (x,s,y,w) of F satisfying $s>0$ and $w>0$ approximates a solution to problem (NIPs), with the approximation becoming increasingly accurate as both $\mu^P(y-y^E)\to 0$ and $\mu^B(w-w^E)\to 0$. If $v=(x,s,y,w)$ is a given approximate zero of F such that $s+\mu^B e>0$ and $w>0$, the Newton equations for the change in variables $d=(\Delta x,\Delta s,\Delta y,\Delta w)$ are given by $F'(v)d=-F(v)$, i.e.,

$$\begin{pmatrix} H(x,y) & 0 & -J(x)^{\mathrm{T}} & 0 \\ 0 & 0 & I & -I \\ J(x) & -I & \mu^P I & 0 \\ 0 & W & 0 & S+\mu^B I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} g(x)-J(x)^{\mathrm{T}}y \\ y-w \\ c(x)-s+\mu^P(y-y^E) \\ s\cdot w+\mu^B(w-w^E) \end{pmatrix}.$$

These equations are identical to the modified-Newton equations (4.8) for minimizing M when $\widehat{H}=H(x,y)$. Theorem 4.1.2 shows that the choice $\widehat{H}=H(x,y)$ is allowed in the neighborhood of a solution satisfying certain second-order optimality conditions, and it follows that the modified-Newton direction used in the proposed method is equivalent asymptotically to the shifted primal-dual path-following directions.

At this point, it is helpful to define some notation that will be useful for later parts of the

chapter.

$$H^M = \begin{pmatrix} H + 2J^{\mathrm{T}} D_P^{-1} J & -2J^{\mathrm{T}} D_P^{-1} & J^{\mathrm{T}} & 0 \\ -2D_P^{-1} J & 2(D_P^{-1} + D_B^{-1}) & -I & I \\ J & -I & D_P & 0 \\ 0 & I & 0 & D_B \end{pmatrix} \approx \nabla^2 M,$$

$$\bar{g} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \quad \bar{H} = \begin{pmatrix} H & 0 \\ 0 & 0 \end{pmatrix}, \quad \bar{J} = \begin{pmatrix} J & -I \\ 0 & I \end{pmatrix}, \quad \bar{D} = \begin{pmatrix} D_P & 0 \\ 0 & D_B \end{pmatrix}$$

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

$$M_j = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}, \quad N_j = \begin{pmatrix} D_P & 0 \\ 0 & D_B \end{pmatrix},$$

$$H^M = \begin{pmatrix} \bar{H} + 2\bar{J}\bar{D}^{-1}\bar{J} & \bar{J}^{\mathrm{T}} \\ \bar{J} & \bar{D} \end{pmatrix}, \quad T_j = \operatorname{diag}(M_j, N_j) = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & D_P & 0 \\ 0 & 0 & 0 & D_B \end{pmatrix}.$$

4.2 A Line-Search Modified Newton Method

In Gill, Kungurtsev & Robinson [13], a line-search algorithm was proposed to minimize the shifted primal-dual penalty-barrier function. The inner iteration minimizes the merit function with fixed penalty and barrier parameters and multiplier estimates.

Three assumptions are required to prove the convergence of the inner iterations, and they are given as follows:

Assumption 4.2.1. *The functions f and c are twice continuously differentiable.*

Algorithm 4.1. Minimizing M for fixed parameters y^E, w^E, μ^P, and μ^B (line search).

procedure MERIT-LS(x_0, s_0, y_0, w_0)
  Restrictions: $s_0 + \mu^B e > 0$, $w_0 > 0$ and $w^E > 0$;
  Constants: $\{\eta, \gamma\} \in (0,1)$;
  Set $v_0 \leftarrow (x_0, s_0, y_0, w_0)$;
  while $\|\nabla M(v_k)\| > 0$ **do**
    Choose $\widehat{H}_k^M \succ 0$, and then compute the search direction d_k from (4.5);
    Set $\alpha_k \leftarrow 1$;
    loop
      if $s_k + \alpha_k \Delta s_k + \mu^B e > 0$ and $w_k + \alpha_k \Delta w_k > 0$ **then**
        if $M(v_k + \alpha_k d_k) \leq M(v_k) + \eta\, \alpha_k \nabla M(v_k)^{\mathrm{T}} d_k$ **then** break;
      end if
      Set $\alpha_k \leftarrow \gamma \alpha_k$;
    end loop
    Set $v_{k+1} \leftarrow v_k + \alpha_k d_k$;
    Set $\widehat{s}_{k+1} \leftarrow c(x_{k+1}) - \mu^P \big(y^E + \frac{1}{2}(w_{k+1} - y_{k+1})\big)$;
    Perform a slack reset $s_{k+1} \leftarrow \max\{s_{k+1}, \widehat{s}_{k+1}\}$;
    Set $v_{k+1} \leftarrow (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$;
  end while
end procedure

Assumption 4.2.2. *The sequence of matrices* $\{H_k^M\}_{k \geq 0}$ *used in* (4.5) *are chosen to be uniformly positive definite and bounded in norm.*

Assumption 4.2.3. *The sequence of iterates* $\{x_k\}$ *is contained in a bounded set.*

The main convergence result is:

Theorem 4.2.1 (Gill, Kungurtsev & Robinson [13])**.** *Under* Assumptions 4.2.1–4.2.3*, the sequence of iterates* $\{v_k\}$ *satisfies* $\lim_{k \to \infty} \nabla M(v_k) = 0$.

In the outer iteration, the algorithm defines quantities to measure the proximity to feasibility, stationarity, and complementarity.

$$\chi_{\text{feas}}(v_{k+1}) = \|c(x_{k+1}) - s_{k+1}\|,$$

$$\chi_{\text{stny}}(v_{k+1}) = \max\big(\|g(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} y_{k+1}\|, \|y_{k+1} - w_{k+1}\|\big), \text{ and}$$

$$\chi_{\text{comp}}(v_{k+1}, \mu_k^B) = \|\min\big(q_1(v_{k+1}), q_2(v_{k+1}, \mu_k^B)\big)\|,$$

where

$$q_1(v_{k+1}) = \max\big(|\min(s_{k+1}, w_{k+1}, 0)|, |s_{k+1} \cdot w_{k+1}|\big) \text{ and}$$
$$q_2(v_{k+1}, \mu_k^B) = \max\big(\mu_k^B e, |\min(s_{k+1} + \mu_k^B e, w_{k+1}, 0)|, |(s_{k+1} + \mu_k^B e) \cdot w_{k+1}|\big).$$

A first-order KKT point v_{k+1} for problem (NIPs) satisfies $\chi(v_{k+1}, \mu_k^B) = 0$, where

$$\chi(v, \mu) = \chi_{\text{feas}}(v) + \chi_{\text{stny}}(v) + \chi_{\text{comp}}(v, \mu). \tag{4.12}$$

With these definitions in hand, the kth iteration is designated as an O-iteration if $\chi(v_{k+1}, \mu_k^B) \leq \chi_k^{\max}$, where $\{\chi_k^{\max}\}$ is a monotonically decreasing positive sequence. The point v_{k+1} is called an O-iterate.

If the condition for an O-iteration does not hold, a test is made to determine if $v_{k+1} = (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$ is an approximate first-order solution of the problem

$$\underset{v=(x,s,y,w)}{\text{minimize}} \; M(v; y_k^E, w_k^E, \mu_k^P, \mu_k^B). \tag{4.13}$$

In particular, the kth iteration is called an M-iteration if v_{k+1} satisfies

$$\|\nabla_x M(v_{k+1}; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \leq \tau_k, \tag{4.14a}$$

$$\|\nabla_s M(v_{k+1}; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \leq \tau_k, \tag{4.14b}$$

$$\|\nabla_y M(v_{k+1}; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \leq \tau_k \|D_{k+1}^P\|_\infty, \text{ and} \tag{4.14c}$$

$$\|\nabla_w M(v_{k+1}; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \leq \tau_k \|D_{k+1}^B\|_\infty, \tag{4.14d}$$

where τ_k is a positive tolerance, $D_{k+1}^P = \mu_k^P I$, and $D_{k+1}^B = (S_{k+1} + \mu_k^B I) W_{k+1}^{-1}$. (See Lemma 4.2.2 for a justification of (4.14).) In this case v_{k+1} is called an M-iterate because it is an approximate first-order solution of (4.13). The multiplier estimates y_{k+1}^E and w_{k+1}^E are defined by the safeguarded values

$$y_{k+1}^E = \max\big(-y_{\max} e, \min(y_{k+1}, y_{\max} e)\big) \text{ and } w_{k+1}^E = \min(w_{k+1}, w_{\max} e) \tag{4.15}$$

for some positive constants $y_{\max}$ and $w_{\max}$.

An iteration that is not an O- or M-iteration is called an F-iteration. In an F-iteration none of the merit function parameters are changed, so that progress is measured solely in terms of the reduction in the merit function.

Algorithm 4.2. A shifted primal-dual penalty-barrier method.

procedure PDB(x_0, s_0, y_0, w_0)
  Restrictions: $s_0 > 0$ and $w_0 > 0$;
  Constants: $\{\eta, \gamma\} \subset (0,1)$ and $\{y_{\max}, w_{\max}\} \subset (0,\infty)$;
  Choose y_0^E, $w_0^E > 0$; $\chi_0^{\max} > 0$; $\tau_0 > 0$; and $\{\mu_0^P, \mu_0^B\} \subset (0,\infty)$;
  Set $v_0 = (x_0, s_0, y_0, w_0)$; $k \leftarrow 0$;
  while $\|\nabla M(v_k)\| > 0$ **do**
    $(y^E, w^E, \mu^P, \mu^B) \leftarrow (y_k^E, w_k^E, \mu_k^P, \mu_k^B)$;
    Compute $v_{k+1} = (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$ in Steps 6–17 of Algorithm 4.1;
    if $\chi(v_{k+1}, \mu_k^B) \le \chi_k^{\max}$ **then** [O-iterate]
      $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\frac{1}{2}\chi_k^{\max}, y_{k+1}, w_{k+1}, \mu_k^P, \mu_k^B, \tau_k)$;
    else if v_{k+1} satisfies (4.14) **then** [M-iterate]
      Set $(\chi_{k+1}^{\max}, \tau_{k+1}) = (\chi_k^{\max}, \frac{1}{2}\tau_k)$; Set y_{k+1}^E and w_{k+1}^E using (4.15);
      if $\chi_{\text{feas}}(v_{k+1}) \le \tau_k$ **then** $\mu_{k+1}^P \leftarrow \mu_k^P$ **else** $\mu_{k+1}^P \leftarrow \frac{1}{2}\mu_k^P$ **end if**
      if $\chi_{\text{comp}}(v_{k+1}, \mu_k^B) \le \tau_k$ and $s_{k+1} \ge -\tau_k e$ **then**
        $\mu_{k+1}^B \leftarrow \mu_k^B$;
      else
        $\mu_{k+1}^B \leftarrow \frac{1}{2}\mu_k^B$; Reset s_{k+1} so that $s_{k+1} + \mu_{k+1}^B e > 0$;
      end if
    else [F-iterate]
      $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\chi_k^{\max}, y_k^E, w_k^E, \mu_k^P, \mu_k^B, \tau_k)$;
    end if
  end while
end procedure

Define

$$\mathscr{O} = \{k : \text{iteration } k \text{ is an O-iteration}\},$$

$$\mathscr{M} = \{k : \text{iteration } k \text{ is an M-iteration}\}, \text{ and}$$

$$\mathscr{F} = \{k : \text{iteration } k \text{ is an F-iteration}\}.$$

The following lemma justifies the substitution used at the beginning of the chapter:

Lemma 4.2.2 (Gill, Kungurtsev & Robinson [13]). *If* $|\mathscr{M}| = \infty$ *then*

$$\lim_{k\in\mathscr{M}} |\pi^Y_{k+1} - y_{k+1}| = \lim_{k\in\mathscr{M}} |\pi^W_{k+1} - w_{k+1}| = \lim_{k\in\mathscr{M}} |\pi^Y_{k+1} - \pi^W_{k+1}| = \lim_{k\in\mathscr{M}} |y_{k+1} - w_{k+1}| = 0.$$

Convergence of the iterates is established using the properties of the *complementary approximate KKT* (CAKKT) *condition* proposed by Andreani, Martínez and Svaiter [1], as described next.

Definition 4.2.1 (CAKKT condition). *A feasible point* (x^*, s^*) *(i.e., a point such that* $s^* \geq 0$ *and* $c(x^*) - s^* = 0$*) is said to satisfy the* CAKKT condition *if there exists a sequence* $\{(x_j, s_j, u_j, z_j)\}$ *with* $\{x_j\} \to x^*$ *and* $\{s_j\} \to s^*$ *such that*

$$\{g(x_j) - J(x_j)^{\mathrm{T}} u_j\} \to 0, \tag{4.16}$$

$$\{u_j - z_j\} \to 0, \tag{4.17}$$

$$\{z_j\} \geq 0, \text{ and} \tag{4.18}$$

$$\{z_j \cdot s_j\} \to 0. \tag{4.19}$$

Any (x^*, s^*) *satisfying these conditions is called a* CAKKT point.

The main result is given as follows:

Theorem 4.2.3 (Gill, Kungurtsev & Robinson [13]). *Under* Assumptions 4.2.1– 4.2.3*, one of the following occurs.*

(i) $|\mathscr{O}| = \infty$*, limit points of* $\{(x_{k+1}, s_{k+1})\}_{k\in\mathscr{O}}$ *exist, and every such limit point* (x^*, s^*) *is a CAKKT point for problem* (NIPs)*. If, in addition, CAKKT holds at* (x^*, s^*)*, then* (x^*, s^*) *is a KKT point for problem* (NIPs)*.*

(ii) $|\mathscr{O}| < \infty$*,* $|\mathscr{M}| = \infty$*, limit points of* $\{(x_{k+1}, s_{k+1})\}_{k\in\mathscr{M}}$ *exist, and every such limit point* (x^*, s^*) *is an infeasible stationary point.*

4.3 A Trust-Region Modified-Newton Method

In the line-search algorithm, there is a crucial assumption that H^M is positive definite. However, this assumption may not hold at points that are far from the solution. Algorithm 4.3 is proposed to overcome this difficulty.

Algorithm 4.3. Minimizing M for fixed y^E, w^E, μ^P, μ^B (trust-region)

procedure MERIT-TR(x_0, s_0, y_0, w_0)
  Restrictions: $s_0 + \mu^B e > 0$, $w_0 > 0$, $w^E > 0$;
  Choose constants $0 < \eta_A < \eta_E < 1$, $\gamma \in (0,1)$, $\{\bar{\gamma}, \nu\} \subset (1,\infty)$, and $\gamma\nu < 1$.
  Set $v_0 \leftarrow (x_0, s_0, y_0, w_0)$;
  while not converged **do**
    Compute an approximate solution $d_k = (\Delta x_k, \Delta s_k, \Delta y_k, \Delta w_k)$ of the problem
    $\min_d \{\nabla M(v_k)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} H^M d : \|d\|_{T_k} \le \delta_j\}$;
    $\rho_k \leftarrow \big(M(v_k + d_k) - M(v_k)\big) / \big(q_k(v_k + d_k) - q_k(v_k)\big)$;
    if $\rho_k \ge \eta_A$ **then**
      Successful iteration $v_{k+1} \leftarrow v_k + d_k$;
      if $\rho_k \ge \eta_E$ **then**
        Set $\delta_k \leftarrow \max\{\delta_k, \bar{\gamma}\|d_k\|_{T_k}\}$;
      else
        $\delta_{k+1} \leftarrow \delta_k$;
      end if
    else
      $v_{k+1} \leftarrow v_k$, $\delta_{k+1} \leftarrow \gamma\|d_k\|_{T_k}$;
    end if
  end while
  Set $\widehat{s}_{k+1} \leftarrow c(x_{k+1}) - \mu^P\big(y^E + \frac{1}{2}(w_{k+1} - y_{k+1})\big)$;
  Reset $s_{k+1} \leftarrow \max\{s_{k+1}, \widehat{s}_{k+1}\}$;
  Set $v_{j+1} = (x_{k+1}, s_{k+1}, y_{k+1})$;
end procedure

The trust region equation can actually be shrunk into a 2 by 2 block matrix, i.e., the equation $(H^M + \sigma T)p = -\nabla M$ can be written in the form

$$\begin{pmatrix} \bar{H} + \sigma I & -\bar{J}^T \\ \bar{J} & \bar{\sigma}\bar{D} \end{pmatrix} \begin{pmatrix} \Delta\bar{x} \\ \Delta\widehat{y} \end{pmatrix} = - \begin{pmatrix} \bar{g} - \bar{J}^{\mathrm{T}}\bar{y} \\ \bar{D}(\bar{y} - \bar{\pi}) \end{pmatrix}, \tag{4.20}$$

where $\bar{\sigma} = (1+\sigma)/(1+2\sigma)$ and $\Delta\widehat{y} = (1+2\sigma)\Delta\bar{y}$. the unsymmetric equations (4.20) may be

Algorithm 4.4. Shifted primal-dual penalty-barrier with a trust region

procedure PDBTR(x_0, s_0, y_0, w_0)
  Choose y_0^E, $w_0^E > 0$, $\chi_0^{\max} > 0$, μ_0^P, $\mu_0^B > 0$;
  Set $v_0 = (x_0, s_0, y_0, w_0)$;
  $(y^E, w^E, \mu^P, \mu^B) \leftarrow (y_k^E, w_k^E, \mu_k^P, \mu_k^B)$;
  while not converged **do**
    Compute $v_{k+1} = (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$ using Algorithm 4.3;
    if $\chi(v_{k+1}, \mu_k^B) \leq \chi_k^{\max}$ **then**; [O-iterate]
      Set $(y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B) \leftarrow (y_{k+1}, w_{k+1}, \mu_k^P, \mu_k^B)$;
      Set $(\chi_{k+1}^{\max}, \tau_{k+1}) \leftarrow (\frac{1}{2}\chi_k^{\max}, \tau_k)$
    else if v_{k+1} satisfies certain condition for ∇M **then** [M-iterate]
      Set $(\chi_{k+1}^{\max}, \tau_{k+1}) \leftarrow (\chi_k^{\max}, \frac{1}{2}\tau_k)$
      Replace y_{k+1}^E, w_{k+1}^E by safeguarded values
      if $\chi_{\text{feas}}(v_{k+1}) \leq \tau_k$ **then**
        $\mu_{k+1}^P \leftarrow \mu_k^P$;
      else
        $\mu_{k+1}^P \leftarrow \frac{1}{2}\mu_k^P$;
      end if
      if $\chi_{\text{comp}}(v_{k+1}, \mu_k^B) \leq \tau_k$ and $s_{k+1} \geq -\tau_k e$ **then**
        Set $\mu_{k+1}^B \leftarrow \mu_k^B$;
      else
        Set $\mu_{k+1}^B \leftarrow \frac{1}{2}\mu_k^B$;
        Reset s_{k+1} so that $s_{k+1} + \mu_{k+1}^B e > 0$;
      end if
    else [F-iterate]
      Set $(y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B) \leftarrow (y_{k+1}, w_{k+1}, \mu_k^P, \mu_k^B)$
      Set $(\chi_{k+1}^{\max}, \tau_{k+1}) \leftarrow (\chi_k^{\max}, \tau_k)$
    end if
  end while
end procedure

written as

$$\begin{pmatrix} H+\sigma I & 0 & -J^T & 0 \\ 0 & \sigma I & I & -I \\ J & -I & \bar{\sigma} D_P & 0 \\ 0 & I & 0 & \bar{\sigma} D_B \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta \widehat{y} \\ \Delta \widehat{w} \end{pmatrix} = - \begin{pmatrix} g - J^{\mathrm{T}} y \\ y - w \\ D_P(y - \pi^Y) \\ D_B(w - \pi^W) \end{pmatrix}, \tag{4.21}$$

where $\bar{\sigma} = (1+\sigma)/(1+2\sigma)$. Using block elimination, the solution of these equations is given by

$$\Delta w = \big(I + \sigma\bar{\sigma} D_B\big)^{-1} \big(y + \Delta y - w - \sigma W^{-1}(s + \mu^B(w - w^E))\big) \tag{4.22}$$

and

$$\Delta s = -W^{-1}\big(s \cdot (w + \bar{\sigma}\Delta w) + \mu^B(w + \bar{\sigma}\Delta w - w^E)\big), \tag{4.23}$$

where Δx and Δy satisfy the equations

$$\begin{pmatrix} H+\sigma I & -J^T \\ J & D_P + \widehat{D}_B \end{pmatrix} \begin{pmatrix} \Delta \widehat{x} \\ \Delta \widehat{y} \end{pmatrix} = - \begin{pmatrix} g - J^{\mathrm{T}} y \\ \widehat{D}_B\big(y - w - \sigma D_B(w - w^E)\big) + D_P\big(y - \pi^Y\big) + D_B\big(w - \pi^W\big) \end{pmatrix}, \tag{4.24}$$

with $\widehat{D}_B = \bar{\sigma} D_B (I + \sigma\bar{\sigma} D_B)^{-1}$.

To prove some boundedness properties in Lemma 4.3.2, the auxiliary function below is needed:

$$\phi_i(x; w_i, w_i^E, \mu^B) = -\mu^B w_i^E \ln x - \mu^B w_i^E \ln(w_i x) + w_i x, \tag{4.25}$$

where $w_i > 0$, $w_i^E > 0$ and $\mu^B > 0$ are fixed constants. For brevity, this function may just be denoted by $\phi(x)$.

Lemma 4.3.1. *The function $\phi_i(x)$ is bounded from below on $(0, \infty)$.*

Proof. Notice ϕ_i is the sum of three convex functions and is thus convex. It is obvious that

$$\lim_{x\to 0+} \phi_i(x) = \lim_{x\to\infty} \phi_i(x) = +\infty.$$

Also

$$\phi'(x) = -2\mu^B w_i^E \frac{1}{x} + w_i,$$

and hence the unique minimum is attained at

$$x = \frac{2}{w_i}\mu^B w_i^E.$$

□

A proof similar to Lemma 4.2 in Gertz & Gill [12] is given below. Here the indices j for the j-th iteration are omitted.

Lemma 4.3.2. *Assume $c_i(x)$ and s_i generated by the algorithm are always bounded from above, and $f(x)$ is always bounded from below by, say, $\bar{f}$. Then the following holds.*

(i) *Each component of $s+\mu^B e$ is bounded above and bounded away from zero.*

(ii) *$w\cdot\left(s+\mu^B e\right)$ is bounded away from zero and bounded from above.*

(iii) *w is bounded above and bounded away from zero.*

(iv) $\dfrac{s_i+\mu^B}{w_i} = \Theta\big((s_i+\mu^B)^2\big)$.

(v) *The eigenvalues of T_k are all positive, bounded from above and away from zero. Therefore, there exists a positive constant c (dependent only on k) such that $(1/c)||v||_2 \le \|v\|_{T_k} \le c||v||_2$.*

Proof. The definition of an iteration guarantees that w_i, $i = 1, 2, \ldots, m$, is always positive. By our algorithm, the following inequality

$$M(v_0) \geq M(v_j) \geq M(v_{j+1}) \geq \bar{f} - \big(c(x)-s\big)^{\mathrm{T}} y^E + \frac{1}{2\mu^P}\|c(x)-s\|^2 + \sum_{i=1}^{m}\Big(-\mu^B w_i^E \ln(s_i+\mu^B) - \mu^B w_i^E \ln\big(w_i(s_i+\mu^B)\big) + w_i(s_i+\mu^B)\Big). \quad (4.26)$$

holds, and may be written as

$$M(v_0) \geq \bar{f} - \big(c(x)-s\big)^{\mathrm{T}} y^E + \frac{1}{2\mu^P}||c(x)-s||^2 + \sum_{i=1}^{m}\phi(s_i+\mu^B)$$

On the right-hand side, $\bar{f}$ and $\sum_{i=1}^{m}\phi(s_i+\mu^B)$ are bounded from below, so $\|c(x)-s\|$ must be bounded, otherwise the right hand side would go to infinity.

1. Now $s_i+\mu^B$ has to be bounded above and bounded below by zero for all i because otherwise, from the preceding lemma, the right-hand side of (4.26) would go to infinity.

2. Now (4.26) can be rewritten as

$$M(v_0) \geq \bar{f} - \big(c(x)-s\big)^{\mathrm{T}} y^E + \frac{1}{2\mu^P}\|c(x)-s\|^2 - \sum_{i=1}^{m}\mu^B w_i^E \ln(s_i+\mu^B) + \sum_{i=1}^{m}\Big(-\mu^B w_i^E \ln\big(w_i(s_i+\mu^B)\big) + w_i(s_i+\mu^B)\Big).$$

   It has been shown that every term except the last is bounded below. If any $s_i+\mu^B$ is not bounded from below by zero and bounded away from zero, the last term would become infinitely large. It follows that $w\cdot(s+\mu^B e)$ is also bounded from below by zero and bounded above.

3. Suppose $0 < m_1 \leq w_i(s_i+\mu^B) \leq M_1$, $0 < m_2 \leq w_i \leq M_2$. Take the quotient of $w_i\big(s_i+\mu^B\big)$ and $s_i+\mu^B e$, the inequality

$$0 < \frac{m_1}{M_2} \leq w_i \leq \frac{M_1}{m_2},$$

   which shows w_i is also bounded above and bounded away from zero.

4. $$\frac{s_i+\mu^B}{w_i} = \frac{(s_i+\mu^B)^2}{w_i(s_i+\mu^B)} = \Theta\big((s_i+\mu^B)^2\big),$$

where the last equality uses the fact that $w_i(s_i + \mu^B)$ is bounded above and away from zero.

5. The last one follows from previous parts. This last result is important, since it shows that the diagonal matrix D_B is bounded away from zero and bounded from above. Since D_P is a constant matrix, it means that T_j-norm and 2-norm are always equivalent. That is to say, there exists $c_1 > c_2 > 0$, such that, for all iterations j,

$$c_1||\cdot||_{T_j} \geq ||\cdot||_2 \geq c_2||\cdot||_{T_j}.$$

Define

$$\widehat{g} = T_j^{-\frac{1}{2}} g, \quad \widehat{B} = T_j^{-\frac{1}{2}} H^M T_j^{-\frac{1}{2}}, \quad \widehat{d} = T_j^{\frac{1}{2}} d,$$

then the 2-norms of these terms are bounded above by a constant multiple of T_j-norms and below by another constant multiple of T_j-norms. Those two constants can be chosen so that they are independent of j.

□

Now the optimization problem becomes

$$\min_{s\in\mathbb{R}^{m+n}} \widehat{g}_j^{\mathrm{T}}\widehat{d} + \frac{1}{2}\widehat{d}^{\mathrm{T}}\widehat{B}_j\widehat{d} \quad \text{s.t. } ||\widehat{d}||_2 \leq \delta_j.$$

Notice

$$||\widehat{g}_j|| = ||g_j||_{T_j} \geq \frac{1}{c_1}||g_j||,$$
$$||B_j|| = \sup_{x\neq 0}\frac{x^{\mathrm{T}}B_j x}{x^{\mathrm{T}}x} = \sup_{x\neq 0}\frac{x^{\mathrm{T}}H^M x}{x^{\mathrm{T}}T_j x} = \sup_{x\neq 0}\frac{x^{\mathrm{T}}Bx}{||x||_{T_j}^2} \leq \sup_{x\neq 0}\frac{||x^{\mathrm{T}}B_j x||}{\frac{1}{c_1^2}||x||^2} = c_1^2||B_j||.$$

Therefore, if $||g_j||$ is bounded from below and $||H^M||$ is bounded from above, so are $||\widehat{g}_j||$ and $||B_j||$, respectively. And hence

$$||\widehat{g}_j||/||B_j|| \geq \frac{1}{c_1^2 c_2}\cdot\frac{||g_j||}{||B_j||}$$

is bounded from below.

Theorem 4.3.3. *If the iteration does not terminate, and all the assumptions are satisfied, we will have* $\liminf \|g_k\| = 0$.

Proof. Suppose not, and we have for some $\varepsilon > 0$, we always have $\|g_k\| > \varepsilon$. Notice that

$$\sum_{l=0}^{j} M(v_l) - M(v_{l+1}) = M(v_0) - M(v_{j+1}).$$

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

$$\mathscr{S} = \Big\{ j \mid M(v_j) - M(v_{j+1}) \geq -\eta_1 q_j(d_j) \Big\}.$$

Summing over all successful iterations, we obtain the inequality

$$\sum_{j \in \mathscr{S}} M(v_j) - M(v_{j+1}) \geq \tau \eta_1 \sum_{j \in \mathscr{S}} ||\widehat{g}_j|| \min\big(\delta_j, ||\widehat{g}_j||/||B_j||\big),$$

which implies that

$$\sum_{j \in \mathscr{S}} \delta_j < \infty.$$

For any longest sequence of unsuccessful iterations, say from $i+1$ to k, the inequality

$$\sum_{j=i+1}^{k} \delta_j \leq \sum_{j=i+1}^{k} \gamma_2^{j-i-1} \gamma_3 \delta_i \leq \frac{\gamma_3}{1-\gamma_2} \delta_i$$

holds. Combining the above two gives

$$\sum_{k} \delta_k \leq \Big(1 + \frac{\gamma_3}{1-\gamma_2}\Big) \sum_{k \in \mathscr{S}} \delta_k < \infty.$$

Notice, in each step, $||d_k|| \leq c_1 ||d_k||_{T_j} = c_1 \delta_k$, which implies the convergence of v_k.

The dual norm of 2-norm is the 2-norm itself. By Taylor expansion,

$$\begin{aligned}||M(v_k) - M(v_k + d_k) + q_k(d_k)|| &\leq ||d_j||_{T_j} \max_{0\leq\xi\leq 1} ||\widehat{g}(v_j + \xi d_j) - \widehat{g}(v_j)|| + \frac{1}{2}\delta_j^2 ||\widehat{B}_j|| \\ &\leq \delta_j \max_{0\leq\xi\leq 1} ||\widehat{g}(v_j + \xi d_j) - \widehat{g}(v_j)|| + \frac{1}{2}\delta_j^2 \cdot \frac{1}{c_2}||B_j|| \\ &\leq \delta_j \max_{0\leq\xi\leq 1} \frac{1}{c_2}||g(v_j + \xi d_j) - g(v_j)|| + \frac{\delta_j^2}{2c_2}||B_j|| \\ &= \frac{\delta_j}{c_2} \max_{0\leq\xi\leq 1} ||g(v_j + \xi d_j) - g(v_j)|| + \frac{\delta_j^2}{2c_2}||B_j||.\end{aligned}$$

If both sides are divided by $-q_k(d_k)$, then

$$||\rho_k - 1|| \leq \frac{\delta_j}{c_2||q_k(d_k)||} \max_{0\leq\xi\leq 1} ||g(v_j + \xi d_j) - g(v_j)|| + \frac{1}{2}\frac{\delta_j^2}{c_2||q_k(d_k)||}||B_j||.$$

Recall the Cauchy point condition, actually for sufficiently small δ_k, that

$$||q_k(d_k)|| \geq \kappa\delta_k.$$

Due to the convergence of v_k (and hence uniform convergence), and the fact that $\delta_k \to 0$, the limit

$$\max_{0\leq\xi\leq 1} ||g(v_j + \xi d_j) - g(v_j)|| \to 0$$

holds.

Along with the boundedness of $||B_j||$, this implies $\rho_j - 1 \to 0$, but the algorithm implies if $\rho_k \to 1$, then $\delta_k \not\to 0$, which is a contradiction. □

Lemma 4.3.4. *The sequence of trust-region radii $\{\delta_k\}$ is bounded.*

Proof. If $k \in \mathscr{S}$, then $v_{k+1} = v_k + d_k$. This update and Assumption 4.2.2 imply that $\{d_k\}_{k\in\mathscr{S}}$ is bounded, which combined with Lemma 4.3.2 gives the existence of a positive scalar, call it A, satisfying $\{||d_k||\}_{k\in\mathscr{S}} \leq A$. Combining this fact with Lines 12 and 14 of Algorithm 4.3 gives

$$\delta_{k+1} \leq \max(\delta_k, \bar{\gamma}||d_k||_{T_k}) \geq \max(\delta_k, \bar{\gamma}A), \quad \forall k \in \mathscr{S}. \tag{4.27}$$

Now for a proof by contradiction, suppose that $\{\delta_k\}$ is unbounded, thus allowing us to define ℓ as the first iteration such that $\delta_\ell > \max\{\delta_0, \bar{\gamma}A\}$. Since the trust-region radius is decreased during iterations $k \notin \mathscr{S}$ (See Line 17 in Algorithm 4.3), it must hold that $(\ell - 1) \in \mathscr{S}$. Combining this with (5.12) shows that $\delta_\ell \leq \max(\delta_{\ell-1}, \bar{\gamma}A)$. Combining this inequality with $\delta_{\ell-1} \leq \max\{\delta_0, \bar{\gamma}A\}$ (recall that ℓ is the first iteration such that $\delta_\ell > \max\{\delta_0, \bar{\gamma}A\}$) shows that $\delta_\ell \leq \max(\delta_0, \bar{\gamma}A)$, thus reaching a contradiction. $\square$

Chapter 5

Primal-Dual Methods for Constrained Problems with Shifts

In the discussion of interior methods in Chapter 4, it was shown how slack variables can be used to avoid the need to find an initial interior point for the nonlinear inequality constraints of the problem

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

where $c:\mathbb{R}^n\to\mathbb{R}^m$ and $f:\mathbb{R}^n\to\mathbb{R}$ are both smooth. In this chapter an alternative approach is proposed in which the nonlinear constraints remain as inequalities, but are shifted so that an initial interior point is available. In this case additional constraints are imposed to force the optimal shifts to be zero. The shifted problem is given by

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

An appropriate merit function for this problem is given by

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

The gradient and Hessian of M may be written in terms of the auxiliary quantities

$$D_P = \mu^P I, \quad D_B = (S+C+\mu^B I)W^{-1}, \quad D_R = \mu^B (C+S+\mu^B I)^{-2} w^E,$$
$$\pi^Y = \pi^Y(x,s) = y^E - \frac{1}{\mu^P} s, \text{ and } \pi^W = \pi^W(s) = \mu^B (S+C+\mu^B I)^{-1} w^E.$$

Using these definitions, the gradient is given by

$$\nabla M = \begin{pmatrix} g - 2\mu^B J^{\mathrm{T}} (C+S+\mu^B I)^{-1} w^E + J^{\mathrm{T}} w \\ \frac{2}{\mu^P} s + (y - 2y^E) - 2\mu^B (C+S+\mu^B I)^{-1} w^E + w \\ s + \mu^P (y - y^E) \\ -\mu^B W^{-1} w^E + \big(c(x) + s + \mu^B e\big) \end{pmatrix}$$
$$= \begin{pmatrix} g - J^{\mathrm{T}} \big(\pi^W + (\pi^W - w)\big) \\ (y - \pi^Y) - (\pi^Y + \pi^W) + (w - \pi^W) \\ -D_P (\pi^Y - y) \\ -D_B (\pi^W - w) \end{pmatrix},$$

and the Hessian is

$$\nabla^2 M = \begin{pmatrix} H + 2\mu^B J^{\mathrm{T}} w^E (S+J+\mu^B I)^{-2} J & 2\mu^B J^{\mathrm{T}} (C+S+\mu^B I)^{-2} w^E & 0 & J^{\mathrm{T}} \\ 2\mu^B w^E (C+S+\mu^B I)^{-2} J & 2D_P^{-1} + 2\mu^B (C+S+\mu^B I)^{-2} w^E & I & I \\ 0 & I & D_P & 0 \\ J & I & 0 & D_B W^{-1} \Pi^W \end{pmatrix}$$
$$= \begin{pmatrix} H + 2J^{\mathrm{T}} D_R J & 2J^{\mathrm{T}} D_R & 0 & J^{\mathrm{T}} \\ 2D_R J & 2D_P^{-1} + 2D_R & I & I \\ 0 & I & D_P & 0 \\ J & I & 0 & D_B W^{-1} \Pi^W \end{pmatrix},$$

where $H = H\big(x, \pi^W + (\pi^W - w)\big)$. The diagonal matrix D_R may be written as

$$\begin{aligned} D_R &= \mu^B w^E (S+J+\mu^B I)^{-2} = (S+J+\mu^B I)^{-1}\mu^B (S+J+\mu^B I)^{-1} w^E \\ &= (S+J+\mu^B I)^{-1}\Pi^W, \end{aligned}$$

Substituting w for π^W gives the approximation

$$D_R \approx (S+J+\mu^B I)^{-1} W = D_B^{-1}.$$

The approximate Hessian can be written as

$$H^M(v_k) = \begin{pmatrix} H + 2J^{\mathrm{T}} D_B^{-1} J & 2J^{\mathrm{T}} D_B^{-1} & 0. & J^{\mathrm{T}} \\ 2D_B^{-1} J & 2D_P^{-1} + 2D_B^{-1} & I & I. \\ 0 & I & D_P & 0. \\ J & I & 0 & D_B \end{pmatrix},$$

which is a function of v_k. Sometimes it will be denoted by H_k^M for brevity.

$$\bar{g} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \quad \bar{H} = \begin{pmatrix} H & 0 \\ 0 & 0 \end{pmatrix}, \quad \bar{J} = \begin{pmatrix} 0 & I \\ J & I \end{pmatrix}, \quad \bar{D} = \begin{pmatrix} D_P & 0 \\ 0 & D_B \end{pmatrix},$$

and

$$H^M = \begin{pmatrix} \bar{H} + 2\bar{J}^{\mathrm{T}} \bar{D}^{-1} \bar{J} & \bar{J}^{\mathrm{T}} \\ \bar{J} & \bar{D} \end{pmatrix},$$

$$T_k = \mathrm{diag}(M_k, N_k) = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & D_P & 0 \\ 0 & 0 & 0 & D_B \end{pmatrix}.$$

As before, it's assumed all $f(x)$ generated by the inner iterations is bounded below by $\bar{f}$.

5.1 Preliminaries

The proof of Lemma 5.1.1 is independent of whether a trust-region or line-search method is used to minimize the merit function. Boundedness in the lemma depends only on obtaining a decrease of the merit function.

The method includes inner and outer iterations. In the inner iteration, the parameters y^E, w^E, μ^P, μ^B are all fixed. As before, the following assumptions are needed:

- f generated by the inner iteration is bounded below by $\bar{f}$,
- f and c are twice continuously differentiable.

Lemma 5.1.1. *If the merit function is decreased in every inner iteration and the conditions* $c(x)+s+\mu^B e$ *and* $w>0$ *are maintained, then the following hold.*

(i) *s is bounded.*

(ii) $c(x)$ *is bounded above.*

(iii) $c(x)+s+\mu^B e$ *is bounded above and bounded away from zero for each of its component.*

(iv) *The component-wise product* $w\cdot\left(c(x)+s+\mu^B e\right)$ *is bounded away from zero and bounded from above.*

(v) *w is bounded above and away from zero.*

(vi) *y is bounded.*

(vii) $\dfrac{c_i(x)+s_i+\mu^B}{w_i}=\Theta\big((c_i(x)+s_i+\mu^B)^2\big)$.

(viii) *The eigenvalues of* T_k *are all positive, bounded from above and away from zero. Therefore, there exists a positive constant c (only dependent on k) such that* $(1/c)||v||_2\leq||v||_{T_k}\leq c||v||_2$.

(ix) *The sequence* $\{\|H^M(v_k)\|\}$ *is bounded. Therefore, there exists a positive constant* κ_2 *such that* $\|H^M(v_k)\| \leq \kappa_2$ *for all* $k \geq 0$.

Proof. From the iteration, the following inequality

$$M(v_0) \geq M(v_j) \geq M(v_{j+1}) \geq \bar{f} - s^{\mathrm{T}} y^E + \frac{1}{2\mu^P}||s||^2 - \sum_{i=1}^{m} \mu^B w_i^E \ln(c_i(x) + s_i + \mu^B) - \sum_{i=1}^{m} \mu^B w_i^E \ln\left(w_i(c_i(x) + s_i + \mu^B)\right) + \sum_{i=1}^{m} w_i(c_i(x) + s_i + \mu^B). \quad (5.1)$$

holds. It can be written as

$$M(v_0) \geq \bar{f} - s^{\mathrm{T}} y^E + \frac{1}{2\mu^P}||s||^2 + \sum_{i=1}^{m} \phi_i\big(c_i(x) + s_i + \mu^B\big).$$

Lemma 5.1.1 shows that ϕ_i's are all bounded from below, and if s is not bounded, then there is a subsequence such that $||s|| \to \infty$, which is impossible because $\bar{f}$ is independent of s, and the quadratic term $\frac{1}{2\mu^P}||s||^2$ grows much faster than the linear term $-s^{\mathrm{T}} y^E$, which leads to a contradiction. It follows that s is bounded. Hence (i) is proved.

It's already known that s is bounded, it suffices to show $c_i(x) + s_i + \mu^B$ is bounded above and away from zero. However, Lemma 4.3.1 shows otherwise $\phi\left(c_i(x) + s_i + \mu^B\right)$ would go to infinity. This proves (ii).

Equation (5.1) can be rewritten as

$$M(v_0) \geq M(v_j) \geq M(v_{j+1}) \geq \bar{f} - s^{\mathrm{T}} y^E + \frac{1}{2\mu^P}||s||^2 - \sum_{i=1}^{m} \mu^B w_i^E \ln(c_i(x) + s_i + \mu^B) + \sum_{i=1}^{m} \left(\mu^B w_i^E \ln\left(w_i(c_i(x) + s_i + \mu^B)\right) + \sum_{i=1}^{m} w_i(c_i(x) + s_i + \mu^B)\right). \quad (5.2)$$

It's shown every term is bounded above except the last term

$$\sum_{i=1}^{m} \left(\mu^B w_i^E \ln\left(w_i(c_i(x) + s_i + \mu^B)\right) + \sum_{i=1}^{m} w_i(c_i(x) + s_i + \mu^B)\right).$$

Similar to function ϕ, $w_i(c_i(x) + s_i + \mu^B)$ has to be bounded above and away from zero, otherwise

this sum would go to infinity. This proves (iii).

Suppose $0 < m_1 < w_i(c_i(x) + s_i + \mu^B) < M_1$, $0 < m_2 < c_i(x) + s_i + \mu^B)$. Then

$$0 < \frac{m_1}{M_2} \leq w_i \leq \frac{M_1}{m_2},$$

which proves (v). Also, (iv) is an immediate result from (iii) and (v). Hence (vi) holds.

The bound $(c_i(x) + s_i + \mu^B e)/w_i$ follows from the equation

$$\frac{c_i(x) + s_i + \mu^B}{w_i} = \frac{\left(c_i(x) + s_i + \mu^B\right)^2}{w_i\left(c_i(x) + s_i + \mu^B\right)} = \Theta\left((c_i(x) + s_i + \mu^B)^2\right).$$

where the last equality uses the fact that $w_i\left(c_i(x) + s_i + \mu^B\right)$ is bounded above and away from zero.

As T_k is given by

$$T_k = \text{diag}(M_k, N_k) = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & D_P & 0 \\ 0 & 0 & 0 & D_B \end{pmatrix},$$

previous arguments show the term D_B is both bounded from above and bounded away from zero. As all other diagonal terms are constant, it can be concluded that T_j is uniformly bounded above and away from zero for all j. This justifies (viii). More precisely, it means that the T_k-norm and the two-norm are always equivalent, i.e., there exists κ, such that, for all iterations k, then

$$1/\kappa_1 ||\cdot||_2 \geq ||\cdot||_{T_k} \geq \kappa_1 ||\cdot||_{T_k}.$$

(ix) is immediate from assumptions and above results. □

5.2 A Line-Search Method

The algorithm is given in Algorithm 5.1. Consider the merit function with terms labeled as follows:

$$M(x,s,y,w;y^E,w^E,\mu^P,\mu^B) = \underbrace{f(x)}_{(A)}\underbrace{-s^{\mathrm{T}}y^E}_{(B)}\underbrace{+\frac{1}{2\mu^P}||s||^2}_{(C)}$$
$$\underbrace{+\frac{1}{2\mu^P}||s+\mu^P(y-y^E)||^2}_{(D)}\underbrace{+\sum_{i=1}^{m} w_i\big(c_i(x)+s_i+\mu^B\big)}_{(G)}$$
$$\underbrace{-\sum_{i=1}^{m}\mu^B w_i^E \ln(c_i(x)+s_i+\mu^B)}_{(E)}\underbrace{-\sum_{i=1}^{m}\mu^B w_i^E \ln\left(w_i(c_i(x)+s_i+\mu^B)\right)}_{(F)}.$$

Some additional assumptions are needed:

- the matrices H_k^M are uniformly positive definite.
- H_k^M is bounded from above.
- ∇M and c are Lipschitz continuous with constant L (this condition can be omitted if the sequence $\{x_k\}$ is bounded.)

Lemma 5.2.1. *The sequence of iterates $\{v_k\}$ satisfies $M(v_{k+1}) < M(v_k)$ for all k.*

Proof. According to the algorithm, the only possibility that $M(v_k)$ does not decrease, is the slack reset. Notice the $\widehat{s}_{k+1}$ gives a minimizer of the sum of (B), (C), (D) and (G) , and it has no effect on (A). Also, it can only decrease the value of (E) and (F). Hence, a slack reset can only decrease the value of M, which concludes the lemma. □

Due to the fact that $M(v_k)$ is decreasing, Lemma 5.1.1 holds as well.

Lemma 5.2.2. *If there exists a positive scalar ε and a subsequence $\mathscr{S}$ satisfying*

$$||\nabla M(v_k)|| \geq \varepsilon, \quad \forall k \in \mathscr{S},$$

then the following results hold

(i) *The set $\{||d_k||\}$ is uniformly bounded above and bounded away from zero.*

Algorithm 5.1. Line-Search subproblem for fixed y^E, w^E, μ^P, μ^B (line-search for shifts)

procedure LS(x_0, s_0, y_0, w_0)
  Restrictions: $w^E > 0$, $w_0 > 0$, $c + s + \mu^B e > 0$
  Constants : $\{\eta, \gamma\} \subset (0,1)$;
  Set $v_0 \leftarrow (x_0, s_0, y_0, w_0)$;
  while $||M(v_k)|| > 0$ **do**
    Choose $H_k^M \succ 0$, compute $\widehat{H}_k^M d_k = -\nabla M(v_k)$;
    Choose $\alpha_k \leftarrow 1$;
    while true **do**
      if $c(x_k + \alpha_k \Delta x_k) + s_k + \alpha_k \Delta s_k + \mu^B e > 0$, $w_k + \alpha_k \Delta w_k > 0$ **then**
        if $M(v_k + \alpha_k d_k) \leq M(v_k) + \eta\, \alpha_k \nabla M(v_k)^{\mathrm{T}} d_k$ **then**
          Break;
        end if
        Set $\alpha_k \leftarrow \gamma \alpha_k$;
      end if
    end while
    Set $v_{k+1} \leftarrow v_k + \alpha_k d_k$;
    Set $\widehat{s}_{k+1} \leftarrow \mu^P y^E + \frac{1}{2}\mu^P (w - y)$;
    Perform a slack reset $s_{k+1} \leftarrow \max\{s_{k+1}, \widehat{s}_{k+1}\}$;
    Set $v_{k+1} \leftarrow v_k + \alpha_k d_k$;
  end while
end procedure

(ii) *There exists a positive scalar* $\delta > 0$ *such that* $\nabla M(v_k)^{\mathrm{T}} d_k \leq -\delta$ *for all* $k \in \mathscr{S}$.

(iii) *There exists a positive scalar* $\alpha_{\min}$ *such that for all* $k \in \mathscr{S}$*, the Armijo condition of the algorithm is satisfied with all* $\alpha_k \leq \alpha_{\min}$.

Proof. The first part comes from the equation

$$H_k^M d_k = -\nabla M(v_k).$$

Hence,

$$\nabla M(v_k)^{\mathrm{T}} d_k = -d_k^{\mathrm{T}} H_k^M d_k \leq -\lambda_{\min}(H_k^M)||d_k||^2 \leq -\delta,$$

for some $\delta > 0$ because $||d_k||$ is bounded below.

For part 3, a standard result of unconstrained optimization is that the Armijo condition is

satisfied for all

$$\alpha_k = \Omega\Big(\frac{-\nabla M(v_k)^{\mathrm{T}} d_k}{||v_k||^2}\Big),$$

which requires the Lipschitz property of ∇M. □

Lemma 5.2.3. *Under the assumptions, the sequence of iterates* $\{v_k\}$ *satisfies* $\lim_{k\to\infty} \nabla M(v_k) = 0$.

Proof. The proof is done by contradiction. Suppose $||\nabla M(v_k)|| \geq \varepsilon > 0$ for some subsequence $k \in \mathscr{S}$. Because $M(v_k)$ is bounded from below, there is some number $M_{\min}$ such that $\lim_{k\to\infty} M(v_k) = M_{\min} > -\infty$. Then

$$\lim_{k\to\infty} \alpha_k \nabla M(v_k)^{\mathrm{T}} d_k = 0,$$

which means $\alpha_k \to 0$. However, by the algorithm, it can only happen when at least one of the feasibility condition is not satisfied.

If $w_k + \alpha_k \Delta w_k > 0$ is not satisfied. It would lead to a contradiction. Since it's proved w_k is always bounded below, say, by $\eta > 0$, and since d_k is bounded, so is Δw_k. Therefore, for sufficiently small α_k,

$$w_k + \alpha_k \Delta w_k > \eta e - \alpha_k ||d_k|| e > 0.$$

If $c_i(x_k + \alpha_k \Delta x_k) + \alpha_k s_i + \Delta s_i + \mu^B > 0$ is not satisfied for some i. It would lead to a contradiction as well. The Lipschitz property of c shows

$$c_i(x_k + \alpha_k \Delta x_k) + s_i + \Delta s_i + \mu^B = c_i(x_k) + [s_k]_i + \mu^B + \big(c_i(x_k + \alpha_k \Delta x_k) - c_i(x_k)\big) + \alpha_k [\Delta s_k]_i$$

say $c_i + s_i + \mu^B > \eta$ for some $\eta > 0$

$$\geq \eta + \big(c_i(x_k + \alpha_k \Delta x_k) - c_i(x_k)\big) + \alpha_k [\Delta s_k]_i$$

use Lipschitz property

$$\geq \eta - \alpha_k L ||\Delta x_k|| - \alpha_k ||[\Delta s_k]_i||.$$

However, by the boundedness of d_k, this will be greater than zero for sufficiently small α_k. $\square$

5.3 Approximate Solutions of the Trust Region Subproblem

This section concerns a primal-dual path-following method based on an approximate solution of the trust-region subproblem. Consider the functions $Q_k(d)$ and $\widehat{Q}_k(d)$ such that

$$\begin{aligned} Q_k(d) &= \nabla M(v_k)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} H^M(v_k) d, \\ \widehat{Q}_k(d) &= \nabla M(v_k)^{\mathrm{T}} d + \frac{1}{2} \min\{0, d^{\mathrm{T}} H^M(v_k) d\}. \end{aligned}$$

Using the model Q_k, a direction $d_k = (\Delta x_k, \Delta s_k, \Delta y_k, \Delta w_k)$ is then computed as an approximate solution to the trust-region subproblem subproblem

$$\begin{array}{ll} \underset{d \in \mathbb{R}^{n+3m}}{\text{minimize}} & Q_k(d) \\ \text{subject to} & ||d||_{T_k} \leq \delta_k. \end{array} \tag{5.3}$$

where $||d||_{T_k} = (d^{\mathrm{T}} T_k d)^{\frac{1}{2}}$. In particular, d_k satisfies the conditions

$$Q_k(d_k) \leq \min\{d_k^{(1)}, d_k^{(2)}\} \tag{5.4a}$$

$$||d_k||_{T_k} \leq \delta_k, \quad \text{and} \tag{5.4b}$$

$$\text{either } \nabla M(v_k)^{\mathrm{T}} d_k < 0, \text{ or } \nabla M(v_k)^{\mathrm{T}} d_k \leq 0 \text{ and } d_k^{\mathrm{T}} H^M(v_k) d_k < 0, \tag{5.4c}$$

where $d_k^{(1)}$ and $d_k^{(2)}$ are given by

$$d_k^{(1)} = -\frac{1}{2} ||\nabla M(v_k)||_2 \min\Big\{ \delta_k \frac{||\nabla M(v_k)||_2}{||\nabla M(v_k)||_{T_k}}, \frac{||\nabla M(v_k)||_2}{||H^M(v_k)||_2} \Big\}, \tag{5.5}$$

$$d_k^{(2)} = \frac{1}{2} \tau \lambda_{\min}\big(H^M(v_k)\big) \Big(\frac{||d_k||_2}{||d_k||_{T_k}} \Big)^2 \delta_k^2. \tag{5.6}$$

The quantity $d_k^{(1)}$ is the model decrease obtained by minimizing Q_k along the direction $-\nabla M(v_k)$ subject to staying within the trust-region constraint. The quantity $d_k^{(2)}$ is τ times the model decrease obtained by minimizing Q_k along the eigenvector associated with the smallest eigenvalue of $H^M(v_k)$ subject to staying within the trust-region. A proof of this fact is given below.

Lemma 5.3.1. *If d_k is the minimizer of Q_k along $-\nabla M(v_k)$ with $\|d_k\|_{T_k} \leq \delta_k$, then*

$$Q_k(d_k) \leq d_k^{(1)}. \tag{5.7}$$

If H_k^M has negative eigenvalues, and d_k is the minimizer of Q_k along the eigenvector with smallest eigenvalue with $\|d_k\|_{T_k} \leq \delta_k$, then

$$Q_k(d_k) \leq d_k^{(2)}. \tag{5.8}$$

Proof. For brevity, the vector g_k will be used to denote ∇M_k when needed.

(1) If d_k is a multiple of $-\nabla M$, condition (5.7) will be satisfied. In this case, it must be a Cauchy point. Suppose first that $\nabla M(v_k) \neq 0$, and that $d_k = -\alpha \nabla M(v_k)$ for some $\alpha > 0$.

- Suppose first that $g_k^{\mathrm{T}} H_k^M g_k > 0$. To minimize the quadratic function Q_k, by (2.3), the step length has to be $\alpha_k^* = g_k^{\mathrm{T}} g_k / g_k^{\mathrm{T}} H_k^M g_k$. If $\alpha_k^* \leq \delta_k / ||g_k||_{T_k}$, then $||\alpha_k^* g_k||_{T_k} \leq \delta_k$, and $\alpha_k = \alpha_k^*$, and the change in the objective is

$$\begin{aligned} Q_k(-\alpha_k^* g_k) &= -\alpha_k^* g_k^{\mathrm{T}} g_k + \frac{1}{2}(\alpha_k^*)^2 g_k^{\mathrm{T}} H_k^M g_k = -\frac{1}{2} g_k^{\mathrm{T}} g_k \Big(\frac{g_k^{\mathrm{T}} g_k}{g_k^{\mathrm{T}} H_k^M g_k} \Big) \leq -\frac{1}{2} \frac{||g_k||_2^4}{||g_k||_2^2 ||H_k^M||_2} \\ &= -\frac{1}{2} \frac{||g_k||_2^2}{||H_k^M||_2}. \end{aligned}$$

  If $\alpha_k^* > \delta_k / ||g_k||_{T_k}$, then $\alpha_k = \delta_k / ||g_k||_{T_k} < \alpha_k^*$, and the change is

$$\begin{aligned} Q_k(-\alpha_k g_k) &= -\alpha_k g_k^{\mathrm{T}} g_k + \frac{1}{2} \alpha_k^2 g_k H_k^M g_k \\ &\leq -\alpha_k g_k^{\mathrm{T}} g_k + \frac{1}{2} \alpha_k \alpha_k^* g_k H_k^M g_k \\ &= -\frac{1}{2} \delta_k \frac{||g_k||_2^2}{||g_k||_{T_k}}. \end{aligned}$$

- Suppose that $g_k^{\mathrm{T}} H_k^M g_k \leq 0$, and there is no minimizer along $-g_k$, and $\alpha_k = \delta_k / ||g_k||_{T_k}$, there is the same bound for the decrease:

$$Q_k(-\alpha_k g_k) = -\alpha_k g_k^{\mathrm{T}} g_k + \frac{1}{2} \alpha_k^2 g_k^{\mathrm{T}} H_k g_k \leq -\alpha_k g_k^{\mathrm{T}} g_k = -\delta_k \frac{||g_k||_2^2}{||g_k||_{T_k}} < -\frac{1}{2} \delta_k \frac{||g_k||_2^2}{||g_k||_{T_k}}.$$

(2) If $\nabla M(v_k) = 0$, condition (5.8) is statisfied. In this case, d_k is an eigenvector associated to the left-most eigenvalue of H_k^M, and $||d_k||_{T_k} = \delta$. In this case the decrease is

$$Q_k(d_k) = \frac{1}{2}\lambda_{\min}\big(H_k^M(v_k)\big)||d_k||_2^2 = \frac{1}{2}\lambda_{\min}\big(H_k^M(v_k)\big)\frac{||d_k||_2^2}{||d_k||_{T_k}^2}\delta_k^2.$$

□

Suppose that f is bounded below by $\bar{f}$ in all the inner iterations. Still define

$$\phi_i(x) = -\mu^B w_i^E \ln x - \mu^B w_i^E \ln(w_i x) + w_i x,$$

as in (4.25), where w_i^E, w_i, μ^B are all positive.

The outer iteration is similar to Algorithm 4.2 and is given as Algorithm 4.4.

The convergence proofs require some assumptions.

Assumption 1: f and c are twice continuously differentiable.

Assumption 2: The sequence of iterates $\{x_k\}$ is contained in a bounded set.

Define the successful iterations

$$\mathscr{S} := \Big\{k \mid \rho_k > \eta_A,\ \ c(x_k + \alpha_k \Delta x_k) + s_k + \Delta s_k + \mu^B e > 0,\ \ w_k + \Delta w_k > 0\Big\}.$$

Lemma 5.3.2. *If $k \notin \mathscr{S}$, then $\delta_{k+1} \leq \gamma\nu\delta_k < \delta_k$.*

Proof. If $k \notin \mathscr{S}$, then $\alpha_k \leq \gamma$, and from the definition of the subproblem, $||d_k||_{T_k} \leq \delta_k$. It follows that $\delta_{k+1} \leq \nu||\alpha_k||_{T_k} = \nu\alpha_k||d_k||_{T_k} \leq \nu\gamma\delta_k < \delta_k$. □

Lemma 5.3.3. *The algorithm is well defined. Moreover, for each iteration $k \geq 0$, it holds that $M(v_{k+1}) \leq M(\widehat{v}_{k+1}) < M(v_k)$.*

Proof. First, consider Line 6 in Algorithm 5.2. As discussed after (5.3), the conditions in Line 6 are satisfied by either the Cauchy point or the eigenpoint as described in [6], which means such an approximate solution is possible.

Algorithm 5.2. Trust-region subproblem for fixed y^E, w^E, μ^P, μ^B (trust-region for shifts)

procedure TR-SHIFTS(x_0, s_0, y_0, w_0)
  Restrictions: $c(x_0) + s_0 + \mu^B e > 0, w_0 > 0, w^E > 0$;
  Choose constants $0 < \eta_A < \eta_E < 1$, $0 < \eta_A < \frac{1}{2}$, $0 < \gamma < 1$, $\{\bar{\gamma}, \nu\} \subset (1, \infty)$ and $\gamma\nu < 1$;
  Set $v_0 \leftarrow (x_0, s_0, y_0, w_0)$;
  while not *converged* **do**
    Compute a direction $d_k = (\Delta x_k, \Delta s_k, \Delta y_k, \Delta w_k)$ satisfying (5.4).
    $\rho_k \leftarrow \big(M(v_k + d_k) - M(v_k)\big)/Q_k(d_k)$;
    if $\rho_k \geq \eta_A$ **then**
      $v_{k+1} \leftarrow v_k + d_k$; [Successful iteration]
      if $\rho_k \geq \eta_E$ **then**
        Set $\delta_k \leftarrow \max\{\delta_k, \bar{\gamma}||d_k||_{T_k}\}$; [Very successful iteration]
      else
        Set $\delta_{k+1} \leftarrow \delta_k$;
      end if
    else
      $\alpha_k = 1$;
      while true **do**
        if $c(x_k + \alpha_k \Delta x_k) + s_k + \alpha_k \Delta s_k + \mu^B e > 0$ and $w_k + \alpha_k \Delta w_k > 0$ **then**
          if $M(v_k + \alpha_k d_k) \leq M(v_k) + \eta_A \widehat{Q}_k(\alpha_k d_k)$ **then**
            Break;
          end if
          Set $\alpha_k \leftarrow \gamma\alpha_k$;
        end if
      end while
      Set $\widehat{v}_{k+1} \leftarrow v_k + \alpha_k d_k$;
      Set $\delta_{k+1} \in \left[\|\alpha_k d_k\|_{T_k}, \nu\|\alpha_k d_k\|_{T_k}\right]$;
    end if
  end while
  Set $\widehat{s}_{k+1} \leftarrow \mu^P[\hat{y}^E - \frac{1}{2}(\hat{y}_k + \hat{w}_k)]$;
  Slack reset $s_{k+1} \leftarrow \max\{s_{k+1}, \widehat{s}_{k+1}\}$;
  Set $v_{k+1} = (\widehat{x}_{k+1}, s_{k+1}, \widehat{y}_{k+1}, \widehat{w}_{k+1})$;
end procedure

Algorithm 5.3. A trust-region primal-dual penalty-barrier method for shifted constraints.

procedure PDBTR-SHIFTS(x_0, s_0, y_0, w_0)
  Restrictions: $s_0 > 0$ and $w_0 > 0$;
  Constants: $\{\eta, \gamma\} \subset (0,1)$ and $\{y_{\max}, w_{\max}\} \subset (0,\infty)$;
  Choose y_0^E, $w_0^E > 0$; $\chi_0^{\max} > 0$; $\tau_0 > 0$; and $\{\mu_0^P, \mu_0^B\} \subset (0,\infty)$;
  Set $v_0 = (x_0, s_0, y_0, w_0)$; $k \leftarrow 0$;
  while $\|\nabla M(v_k)\| > 0$ **do**
    $(y^E, w^E, \mu^P, \mu^B) \leftarrow (y_k^E, w_k^E, \mu_k^P, \mu_k^B)$;
    Compute $v_{k+1} = (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$ by Algorithm 5.2;
    if $\chi(v_{k+1}, \mu_k^B) \le \chi_k^{\max}$ **then** [O-iterate]
      $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\frac{1}{2}\chi_k^{\max}, y_{k+1}, w_{k+1}, \mu_k^P, \mu_k^B, \tau_k)$;
    else if v_{k+1} satisfies (4.14) **then** [M-iterate]
      Set $(\chi_{k+1}^{\max}, \tau_{k+1}) = (\chi_k^{\max}, \frac{1}{2}\tau_k)$; Set y_{k+1}^E and w_{k+1}^E using (4.15);
      if $\chi_{\text{feas}}(v_{k+1}) \le \tau_k$ **then** $\mu_{k+1}^P \leftarrow \mu_k^P$ **else** $\mu_{k+1}^P \leftarrow \frac{1}{2}\mu_k^P$ **end if**
      if $\chi_{\text{comp}}(v_{k+1}, \mu_k^B) \le \tau_k$ and $s_{k+1} \ge -\tau_k e$ **then**
        $\mu_{k+1}^B \leftarrow \mu_k^B$;
      else
        $\mu_{k+1}^B \leftarrow \frac{1}{2}\mu_k^B$; Reset s_{k+1} so that $s_{k+1} + \mu_{k+1}^B e > 0$;
      end if
    else [F-iterate]
      $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\chi_k^{\max}, y_k^E, w_k^E, \mu_k^P, \mu_k^B, \tau_k)$;
    end if
  end while
end procedure

Second, it must be shown that the loop starting in Algorithm 5.2 Line 17 terminates finitely. It is known that finite termination will occur because either $\nabla M(v_k)^{\mathrm{T}} d_k < 0$, or $\nabla M(v_k)^{\mathrm{T}} d_k \le 0$ and $d_k^{\mathrm{T}} H^M(v_k) d_k < 0$. Note that $\widehat{Q}_k(d_k) \le Q_k(d_k) < 0$ for all k. If the update in Algorithm 5.2 Line 7 takes place, then it follows from the inequalities $\rho_k \ge \eta_A$ and $\widehat{Q}_k(d_k) < 0$ that $M(\widehat{v}_{k+1}) < M(v_k)$. If the update in Algorithm 5.2, Line 29, takes place, the line search produces an α_k such that $\widehat{v}_{k+1} = v_k + \alpha_k d_k$ satisfies $M(\widehat{v}_{k+1}) < M(v_k)$. Therefore, regardless of which update is used, the inequality $M(\widehat{v}_{k+1}) < M(v_k)$ holds.

The slack reset uses $\widehat{s}_{k+1}$, which is the minimizer of M with respect to s of the sum of terms (B), (C), (D), (G). Therefore, the sum of these terms cannot increase. Also, (A) is independent of s. In addition the slack reset cannot decrease the value of s, and hence cannot decrease the value of (E) or (F). It follows that $M(v_{k+1}) \le M(\widehat{v}_{k+1})$, as required. □

Lemma 5.3.4. *If there exists some $\varepsilon > 0$, such that $||\nabla M(v_k)||_2 \geq \varepsilon$ for all k, then $\sum_{k=0}^{\infty} \delta_k < \infty$, and the sequence $\{v_k\}$ converges.*

Proof. There are two cases, depending on the cardinality of $\mathscr{S}$.

Case 1: Bounded $|\mathscr{S}|$. In this case there exists some $\bar{k}$ such that $k \notin \mathscr{S}$ for all $k \geq \bar{k}$, and it follows that $\delta_{k+1} < \gamma\nu\delta_k$, and hence $\delta_k \leq (\gamma\nu)^{k-\bar{k}}\delta_{\bar{k}}$. This leads to the bound $\displaystyle\sum_{k=\bar{k}}^{\infty} \delta_k < \infty$, which shows $\displaystyle\sum_{k=1}^{\infty} \delta_k < \infty$.

Case 2: Infinite $|\mathscr{S}|$. Summing over all successful iterations shows

$$\begin{aligned} M(v_0) - M_{\text{low}} &= \sum_{j\in\mathscr{S}} M(v_j) - M(v_{j+1}) \geq \sum_{j\in\mathscr{S}} M(v_j) - M(\widehat{v}_{j+1}) \\ &= \sum_{j\in\mathscr{S}} \big(M(v_j) - M(v_j + d_j)\big) \\ &\geq -\sum_{j\in\mathscr{S}} \eta_A Q_j(d_j) \geq -\sum_{j\in\mathscr{S}} \eta d_k^{(I)} \\ &= \frac{1}{2}\eta_A ||\nabla M(v_k)||_2 \min\left\{ \delta_k \frac{||\nabla M(v_k)||_2}{||\nabla M(v_k)||_{T_k}}, \frac{||\nabla M(v_k)||_2}{||H^M(v_k)||_2} \right\}. \end{aligned} \tag{5.9}$$

As $\dfrac{||\nabla M(v_k)||_2}{||\nabla M(v_k)||_{T_k}}$ is bounded from below and $||\nabla M(v_k)|| \geq \varepsilon$, it follows that

$$\sum_{j\in\mathscr{S}} \delta_j < \infty. \tag{5.10}$$

Let $\{k_\ell\}_{\ell=1}^{\infty}$ denote the infinite subsequence of iterations consisting of only the successful iterations. Therefore, for each $\ell \geq 1$, it holds that

$$k_\ell \in \mathscr{S}, \quad k_{\ell+1} \in \mathscr{S}, \quad \text{and} \quad j \notin \mathscr{S} \quad \text{for all } j \text{ satisfying} \quad k_\ell < j < k_{\ell+1}.$$

Using these properties, the fact that $\delta_{k_\ell+1} \leq \bar{\gamma}\delta_{k_\ell}$ for all $\ell \geq 1$ by construction of Algorithm 5.2, and Lemma 5.3.2, it follows that

$$\sum_{k=k_\ell}^{k_{\ell+1}-1} \delta_k \leq \delta_{k_\ell} + \sum_{k=k_\ell+1}^{k_{\ell+1}-1} (\gamma\nu)^{k-k_\ell-1}\bar{\gamma}\delta_{k_\ell} \leq \delta_{k_\ell} + \frac{\bar{\gamma}\delta_{k_\ell}}{1-\gamma\nu} = c\delta_{k_\ell}, \tag{5.11}$$

where $c = (1-\gamma\nu+\bar{\gamma})/(1-\gamma\nu) \in (1,\infty)$. It follows from (5.11) and (5.10) that

$$\sum_{k=0}^{\infty}\delta_k = \sum_{k=0}^{k_1-1}\delta_k + \sum_{\ell=1}^{\infty}\left(\sum_{k=k_\ell}^{k_{\ell+1}-1}\delta_k\right) \leq \sum_{k=0}^{k_1-1}\delta_k + c\sum_{\ell=1}^{\infty}\delta_{k_\ell} = \sum_{k=0}^{k_1-1}\delta_k + c\sum_{k\in\mathscr{S}}\delta_k < \infty.$$

Notice from the Algorithm 5.2, we know $(x_k, 0, y_k, w_k)$ is a Cauchy sequence under norm $||\cdot||_{T_k}$. Since $||\cdot||_2$ and $||\cdot||_{T_k}$ are equivalent norms, it's also convergent under the Euclidean norm. Line 29 in Algorithm 5.2 shows s_k is convergent as well. This shows $v_k = (x_k, s_k, y_k, w_k)$ is also convergent, which completes the proof. □

Theorem 5.3.5. *It holds that* $\liminf_{k\to\infty}||g_k|| = 0$.

Proof. The proof is by contradiction. Suppose that there exists $\varepsilon > 0$ and $\bar{k}$ such that

$$||\nabla M(v_k)||_2 \geq \varepsilon,$$

for all $k \geq \bar{k}$. A Taylor-series expansion gives

$$|M(v_k) - M(v_k+d_k) + Q_k(d_k)| \leq |M(v_k) - \left(M(v_k) + \nabla M(v_k)^{\mathrm{T}}d_k\right) - \tfrac{1}{2}d_k^{\mathrm{T}}H_k^M d_k|$$

The remainder theorem implies that there exists $\xi \in (0,1)$ such that

$$\begin{aligned}|M(v_k) - \left(M(v_k) + \nabla M(v_k)^{\mathrm{T}}d_k\right) - \tfrac{1}{2}d_k^{\mathrm{T}}H^M d_k| &\leq \left|\frac{1}{2}d_k^{\mathrm{T}}\nabla^2 M(v_k+\xi_k d_k)d_k - \frac{1}{2}d_k^{\mathrm{T}}H^M d_k\right| \\ &\leq \frac{1}{2}||d_k||_2^2||\nabla^2 M(v_k+\xi_k d_k) - H_k^M||_2\end{aligned}$$

bound by the trust region radius and use the relationship of 2-norm and T_k-norm

$$\leq \frac{1}{2}\kappa_1^2\delta_k^2||\nabla^2 M(v_k+\xi_k d_k) - H_k^M||_2.$$

The Cauchy-Point condition (2.2) states, for sufficiently small δ_k, the inequality

$$||Q_k(d_k)|| \geq \kappa\delta_k.$$

holds for some positive constant κ. Dividing both sides by $-Q_k(d_k)$ gives

$$||\rho_k - 1|| \leq \frac{\kappa_1^2 \delta_k^2 ||\nabla^2 M(v_k + \xi_k d_k) - H_k^M||_2}{2|Q_k(d_k)|} = O(\delta_k).$$

Here, we used the fact that $||\nabla^2 M(v_k + \xi_k d_k) - H_k^M||_2$ is bounded, which comes from the fact that $\delta_k \to 0$ and boundedness proved in Lemma 5.1.1.

As $\delta_k \to 0$, it must hold that $\rho_k \to 1$. This implies that that for k sufficiently large, $k \in \mathscr{S}$. However, in this case $\delta_k \not\to 0$, which is a contradiction. □

Lemma 5.3.6. *The sequence of trust-region radii $\{\delta_k\}$ is bounded.*

Proof. If $k \in \mathscr{S}$, then $v_{k+1} = v_k + d_k$. This update and Assumption 2 imply that $\{d_k\}_{k \in \mathscr{S}}$ is bounded, which combined with Lemma 5.1.1(viii) gives the existence of a positive scalar, call it B, satisfying $\{||d_k||\}_{k \in \mathscr{S}} \leq B$. Combining this fact with Lines 11 and 13 of Algorithm 5.2 gives

$$\delta_{k+1} \leq \max(\delta_k, \bar{\gamma}||d_k||_{T_k}) \geq \max(\delta_k, \bar{\gamma}B), \quad \forall k \in \mathscr{S}. \tag{5.12}$$

Now for a proof by contradiction, suppose that $\{\delta_k\}$ is unbounded, thus allowing us to define ℓ as the first iteration such that $\delta_\ell > \max\{\delta_0, \bar{\gamma}B\}$. Since the trust-region radius is decreased during iterations $k \notin \mathscr{S}$ (See Lemma 5.3.2), it must hold that $(\ell - 1) \in \mathscr{S}$. Combining this with (5.12) shows that $\delta_\ell \leq \max(\delta_{\ell-1}, \bar{\gamma}B)$. Combining this inequality with $\delta_{\ell-1} \leq \max\{\delta_0, \bar{\gamma}B\}$ (recall that ℓ is the first iteration such that $\delta_\ell > \max\{\delta_0, \bar{\gamma}B\}$) shows that $\delta_\ell \leq \max(\delta_0, \bar{\gamma}B)$, thus reaching a contradiction. □

Lemma 5.3.7. *Let $\bar{\alpha} \in (0, \frac{1}{\nu})$. If Algorithm 5.2 returns $\alpha_k \geq \bar{\alpha}$, then*

$$M(v_k) - M(v_k + \alpha_k d_k) \geq -\eta_A \bar{\alpha} \alpha_1 \min\{d_k^{(1)}, d_k^{(2)}\}.$$

Proof. Let k be any iteration such that $\alpha_k > \bar{\alpha}$. Since $\nu \in (1, \infty)$ in Algorithm 5.2, it follows that $\bar{\alpha} \in (0, \frac{1}{\nu}) \subset (0, 1)$. If $d_k^{\mathrm{T}} H_k^M(v_k) d_k \geq 0$, then it follows from the definition of $\widehat{Q}_k$, $\widehat{Q}_k(d_k) < 0$, and $\bar{\alpha} \in (0, 1)$ that

$$\bar{\alpha} \alpha_k \widehat{Q}_k(d_k) \geq \alpha_k \widehat{Q}_k(d_k) = \alpha_k \nabla M(v_k)^{\mathrm{T}} d_k = \nabla M(v_k)^{\mathrm{T}} (\alpha_k d_k) = \widehat{Q}_k(\alpha_k d_k).$$

On the other hand, if $d_k^{\mathrm{T}} H^M(v_k) d_k < 0$, then it follows from the definition of $\widehat{Q}_k$, $\nabla M(v_k)^{\mathrm{T}} d_k \leq 0$, $\bar{\alpha} \in (0,1)$ and $\alpha_k \geq \bar{\alpha}$ that

$$\begin{aligned}\bar{\alpha}\alpha_k \widehat{Q}_k(d_k) &= \bar{\alpha}\alpha_k \nabla M(v_k)^{\mathrm{T}} d_k + \frac{1}{2}\bar{\alpha}\alpha_k d_k^{\mathrm{T}} H_k^M(v_k) d_k \\ &\geq \alpha_k \nabla M(v_k)^{\mathrm{T}} d_k + \frac{1}{2}\alpha_k^2 d_k^{\mathrm{T}} H_k^M(v_k) d_k = \widehat{Q}_k(\alpha_k d_k).\end{aligned}$$

Putting these two cases together shows that $\bar{\alpha}\alpha_k \widehat{Q}_k(d_k) \geq \widehat{Q}_k(\alpha_k d_k)$ for all $k \geq 0$. If the iterate update in Line 25 in Algorithm 5.2 happens, then $M(v_k) - M(v_k + \alpha_k d_k) \geq -\eta_A \widehat{Q}_k(\alpha_k d_k)$, which combined with $\bar{\alpha}\alpha_k \widehat{Q}_k(d_k) \geq \widehat{Q}_k(\alpha_k d_k)$ gives the inequality

$$M(v_k) - M(v_k + \alpha_k d_k) \geq -\eta_A \widehat{Q}_k(d_k) \geq -\eta_A \bar{\alpha}\alpha_k \widehat{Q}_k(d_k) \geq -\eta_A \bar{\alpha}\alpha_k Q_k(d_k).$$

On the other hand, if the iterate update in Line 9 in Algorithm 5.2 occurs, then

$$M(v_k) - M(v_k + \alpha_k d_k) \geq -\eta_A Q_k(d_k) \geq -\eta_A \bar{\alpha}\alpha_k Q_k(d_k).$$

Regardless of which update occurs, $M(v_k) - M(d_k + \alpha_k d_k) \geq -\eta_A \bar{\alpha}\alpha_k Q_k(d_k)$. The desired result follows from this inequality, $\widehat{Q}_k(d_k) \leq Q_k(d_k) \leq 0$ and (5.4). □

5.3.1 The path-following equations

Recall that the approximate Hessian and the gradient of the merit function are

$$H^M = \begin{pmatrix} H + 2J^{\mathrm{T}} D_B^{-1} J & 2J^{\mathrm{T}} D_B^{-1} & 0 & J^{\mathrm{T}} \\ 2D_B^{-1} J & 2D_P^{-1} + 2D_B^{-1} & I & I \\ 0 & I & D_P & 0 \\ J & I & 0 & D_B \end{pmatrix},$$

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

and the approximate Newton equations for minimizing M are

$$\begin{pmatrix} H+2J^{\mathrm{T}}D_B^{-1}J & 2J^{\mathrm{T}}D_B^{-1} & 0 & J^{\mathrm{T}} \\ 2D_B^{-1}J & 2D_P^{-1}+2D_B^{-1} & I & I \\ 0 & I & D_P & 0 \\ J & I & 0 & D_B \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} g - J^{\mathrm{T}}\big(\pi^W + (\pi^W - w)\big) \\ (y-\pi^Y) - (\pi^Y + \pi^W) + (w-\pi^W) \\ -D_P(\pi^Y - y) \\ -D_B(\pi^W - w) \end{pmatrix}. \quad (5.13)$$

Consider the nonsingular upper-triangular matrix

$$U = \begin{pmatrix} I & 0 & 0 & -2J^{\mathrm{T}}D_B^{-1} \\ 0 & I & -2D_P^{-1} & -2D_B^{-1} \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & W \end{pmatrix},$$

then the matrix UH^M and vector $U\nabla M$ are given by

$$\begin{pmatrix} H & 0 & 0 & -J^{\mathrm{T}} \\ 0 & 0 & -I & -I \\ 0 & I & D_P & 0 \\ WJ & W & 0 & (C+S+\mu^B I) \end{pmatrix} \text{ and } \begin{pmatrix} g - J^{\mathrm{T}}w \\ -y-w \\ s+\mu^P(y-y^E) \\ \big(c(x)+s\big)w + \mu^B(w-w^E) \end{pmatrix}. \quad (5.14)$$

These equations can be interpreted in a different way. Let $v^* = (x^*, s^*, y^*, w^*)$ be a KKT point, where w^* and y^* are the Lagrange multipliers associated with the constraints $c(x)+s \geq 0$ and $s = 0$, respectively. The Lagrangian is

$$L = f(x) - \big(c(x)+s\big)^{\mathrm{T}}w - s^{\mathrm{T}}y,$$

and the KKT conditions are

$$
\begin{aligned}
c(x^*) + s^* &\geq 0, && \text{(feasibility)} \\
s^* &= 0 && \text{(feasibility)} \\
\big(c(x^*) + s^*\big) \cdot w^* &= 0, && \text{(complementarity slackness)} \\
w^* &\geq 0, && \text{(sign of the multipliers)} \\
0 = \nabla_x L &= g - J^{\mathrm{T}} w, && \text{(optimality)} \\
0 = \nabla_s L &= -w - y. && \text{(optimality)}
\end{aligned}
$$

These equations can be perturbed so that

$$
\begin{aligned}
c(x^*) + s^* &\geq 0, && \text{(feasibility)} \\
s^* &= -\mu^P (y - y^E), && \text{(feasibility)} \\
\big(c(x^*) + s^*\big) \cdot w^* &= -\mu^B (w - w^E), && \text{(complementarity slackness)} \\
w^* &\geq 0, && \text{(sign of the multipliers)} \\
0 = \nabla_x L &= g - J^{\mathrm{T}} w, && \text{(optimality)} \\
0 = \nabla_s L &= -w - y. && \text{(optimality)}
\end{aligned}
$$

The application of Newton's method for a zero of these equations gives Newton equations that are identical to the equations (5.14) above.

5.4 Form of General Problems

5.4.1 Upper and Lower Bounds on Constraints and Variables

Consider the case where there are both upper and lower bounds on constraints and variables. The problem can be written as

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & l^C \le c(x) \le u^C, \quad l^X \le x \le u^X. \end{aligned}$$

With shift variables, the problem can be rewritten as follows:

$$\begin{aligned} \underset{x\in\mathbb{R}^n; l^X, u^X\in\mathbb{R}^n; l^C, u^C\in\mathbb{R}^m}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) - l^C + s \ge 0, \;\; u^C - c(x) + s \ge 0, \;\; s = 0 \\ & x - l^X + t \ge 0, \;\; u^X - x + t \ge 0, \;\; t = 0. \end{aligned}$$

Here L_L, L_U, L_X are matrices formed from rows of I_m and E_L, E_U, E_X are matrices formed from rows of I_n, so that we have upper bounds, lower bounds, and equalities on constraints and variables respectively.

Appropriate equations can be formulated as described in Chapter 4. The detailed computations are given in Appendix A.

Chapter 6

Numerical Experiments

6.1 The implementation

Numerical results were obtained for the primal-dual trust-region method `pdbtr`. All testing was done on problems taken from the CUTEst test collection (see Bongartz, Conn, Gould and Toint [2] and Gould, Orban and Toint [20]).

Numerical results were obtained for MATLAB implementations of two variants of the shifted interior method. Algorithm `pdb` is the shifted primal-dual method of Gill, Kungurtsev and Robinson [13]. This method uses a modified Newton method with a conventional Armijo line search. Algorithm `pdbtr` is the shifted primal-dual method with the trust-region method discussed above.

Each CUTEst problem may be written in the form

$$\underset{x}{\text{minimize}} \;\; f(x) \quad \text{subject to} \quad \begin{pmatrix} \ell^x \\ \ell^s \end{pmatrix} \leq \begin{pmatrix} x \\ c(x) \end{pmatrix} \leq \begin{pmatrix} u^x \\ u^s \end{pmatrix}, \tag{6.1}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}^m$, $f : \mathbb{R}^n \mapsto \mathbb{R}$, and (ℓ^x, ℓ^s) and (u^x, u^s) are constant vectors of lower and upper bounds. In this format, a fixed variable or an equality constraint has the same value for its upper and lower bounds. A variable or constraint with no upper or lower limit is indicated by a bound of $\pm 10^{20}$. The approximate Newton equations for problem (6.1) are derived in Appendix A. As is the case for problem (NIPs) the principal work at each iteration is the solution of a reduced

$(n+m)\times(n+m)$ KKT system analogous to (5.13). Each KKT matrix was factored using the MATLAB built-in command `LDL`. Exact second derivatives were used for all the runs.

For Algorithm `pdb`, if the KKT matrix is this matrix is singular or has more than m negative eigenvalues, the Hessian of the Lagrangian H is modified using the method of Wächter and Biegler [32, Algorithm IC, p. 36], which factors the KKT matrix with δI_n added to H. At any given iteration the value of δ is increased from zero if necessary until the inertia of the KKT matrix is correct.

The relative performance of the solvers is summarized using performance profiles (in $\log_2$ scale), which were proposed by Dolan and Moré [8]. Let $\mathscr{P}$ denote a set of problems used for a given numerical experiment. For each method s we define the function $\pi_s : [0, r_M] \mapsto \mathbb{R}^+$ such that

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

where n_p is the number of problems in the test set and $r_{p,s}$ denotes the ratio of the number of function evaluations needed to solve problem p with method s and the least number of function evaluations needed to solve problem p. If method s failed for problem p, then $r_{p,s}$ is set to be twice of the maximal ratio. The parameter r_M is the maximum value of $\log_2(r_{p,s})$.

The iterates were terminated at the first point that satisfied the conditions $e_P(x,s) < \tau_P$ and $e_D(x,s,y,w) < \tau_D$, where e_P and e_D are the primal and dual infeasibilities

$$e_P(x,s) = \left\| \begin{pmatrix} \min\left\{ 0, s \right\} \\ \|c(x) - s\|_\infty / \max\left\{ 1, \|s\|_\infty \right\} \end{pmatrix} \right\|_\infty, \tag{6.2a}$$

and

$$e_D(x,s,y,w) = \left\| \begin{pmatrix} \|\nabla f(x) - J(x)^{\mathrm{T}} y\|_\infty / \sigma \\ \|w - y\|_\infty \\ w \cdot \min\left\{ 1, s \right\} \end{pmatrix} \right\|_\infty, \tag{6.2b}$$

with $\sigma = \max\left\{ 1, \|\nabla f(x)\|, \max\left\{ 1, \|y\| \right\} \|J(x)\|_\infty \right\}$. Similarly, the iterates were terminated at

an infeasible stationary point (x,s) if $e_P(x,s) > \tau_P$, $\min\{0, s\} \leq \tau_P$ and $e\mathbb{I}(x,s) \leq \tau_{\text{inf}}$, where

$$e\mathbb{I}(x,s) = \left\| J(x)^{\mathrm{T}}(c(x) - s) \cdot \min\{1, s\} \right\|_\infty / \sigma. \tag{6.3}$$

6.2 Numerical results

The runs were done using MATLAB version R2022b on an iMac Pro with a 3.0 GHz Intel Xeon W processor and 128 GB of 800 MHz DDR4 RAM running macOS, version 10.14.6 (64 bit). Results were obtained for six subsets of problems from the CUTEst test collection. The subsets consisted of 171 problems with no constraints (problems UC); 135 problems with a general nonlinear objective and upper and lower bounds on the variables (problems BC); 212 problems with a general nonlinear objective, general linear constraints and bounds on the variables (problems LC); 124 problems formulated by Hock and Schittkowski ([24]) (problems HS); 372 problems with a general nonlinear objective, general linear and nonlinear constraints and bounds on the variables (problems NC); and 117 problems with a quadratic objective, general linear constraints and bounds on the variables (problems QP). The UC, BC, LC, NC and QP subsets were selected based on the number of variables and general constraints. In particular, a problem was chosen if the associated KKT system was of the order of 1000 or less. The same criterion was used to set the dimension of those problems for which the problem size can be specified. The nonsmooth problem `hs87` was excluded from the Hock-Schittkowski problems.

All the MATLAB implementations were initialized with identical parameter values that were chosen based on the empirical performance on the entire collection of problems. A summary of the values is given in Table 6.1. The initial primal-dual estimate $(x_0,\, y_0)$ was based on the default initial values supplied by CUTEst. If necessary, x_0 was projected onto the set $\{x : \ell^x \leq x \leq u^x\}$ to ensure feasibility with respect to the bounds on x. The iterates were terminated at the first point that satisfied the conditions (6.2a)–(6.2b) or (6.3) defined in terms of the constraints associated with problem (6.1).

Figures 6.1–6.6 present the performance profiles for the total number of iterations and

Table 6.1. Control parameters for Algorithms **pdb** and **pdbtr**.

Parameter	Description	Value
$y_{\max}$, $w_{\max}$	Maximum allowed y^E, w^E	1.0e+6
μ_0^P	Initial penalty parameter for Algorithm 4.4	1.0e-4
μ_0^L	Initial flexible line-search penalty parameter for Algorithm 4.2	1.0
μ_0^B	Initial barrier parameter for Algorithm 4.4	1.0e-4
τ_0	Initial termination tolerance for specifying an M-iterate	0.5
τ_P	Primal feasibility tolerance (6.2a)	1.0e-4
τ_D	Dual feasibility tolerance (6.2b)	1.0e-4
τ_{inf}	Infeasible stationary point tolerance (6.3)	1.0e-4
$\chi_0^{\max}$	Initial target for an O-iteration	1.0e+3
η_A	Line-search Armijo sufficient reduction	1.0e-2
η_F	Line-search sufficient reduction for $\|F\|$	1.0e-2
γ_A	Line-search factor for reducing an Armijo step	1.0e-3
f_{unb}	Unbounded objective	1.0e-9
$k_{\max}$	Iteration limit for all algorithms	500

function evaluations required to solve the 171 UC problems, 135 BC problems, 212 LC problems, 124 HS problems, 372 NC problems, and 115 QP problems successively. The profiles show that the relative performance of trust-region interior method `pdbtr` depends on the problem category. Fewer iterations and function evaluations than `pdb` are required for UC and BC problems, but the performance is mixed on general problems. For each type of problems, the left figures give the profiles for the number of function evaluations, and the right figures give the profiles for the number of iterations.

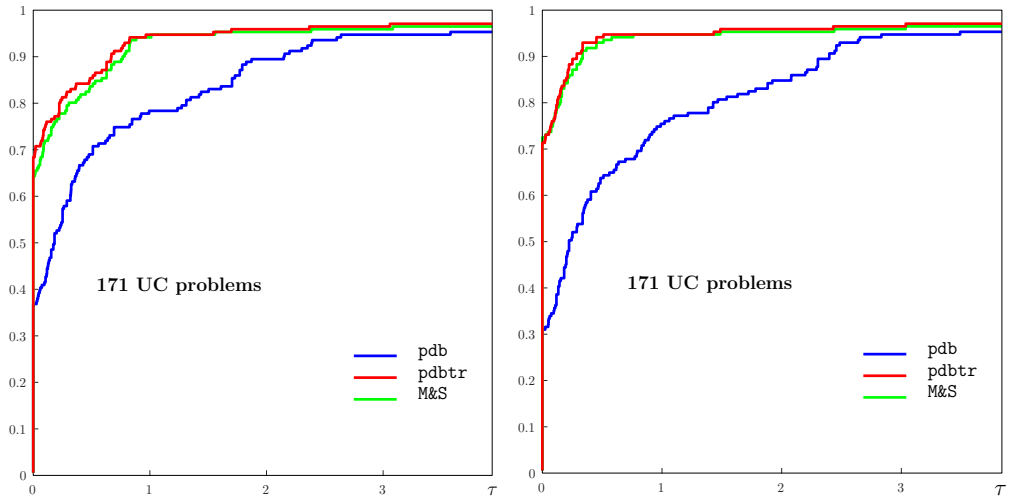

Figure 6.1. Performance profiles for the primal-dual interior algorithms `pdb`, `pdbtr`, and **pdbtrChol** applied to 171 unconstrained (UC) problems from the CUTEst test set.

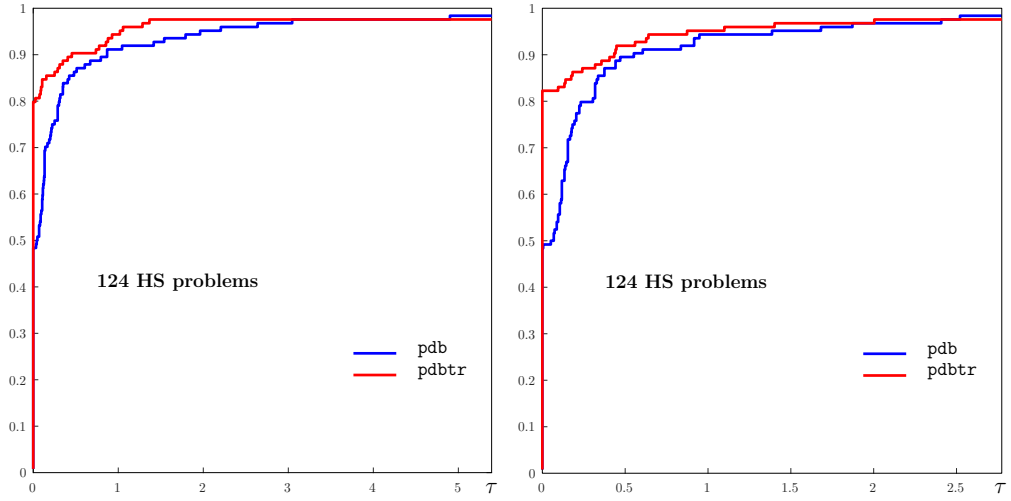

Figure 6.2. Performance profiles for the primal-dual interior algorithms `pdb` and `pdbtr` applied to 124 Hock-Schittkowski (HS) problems from the CUTEst test set.

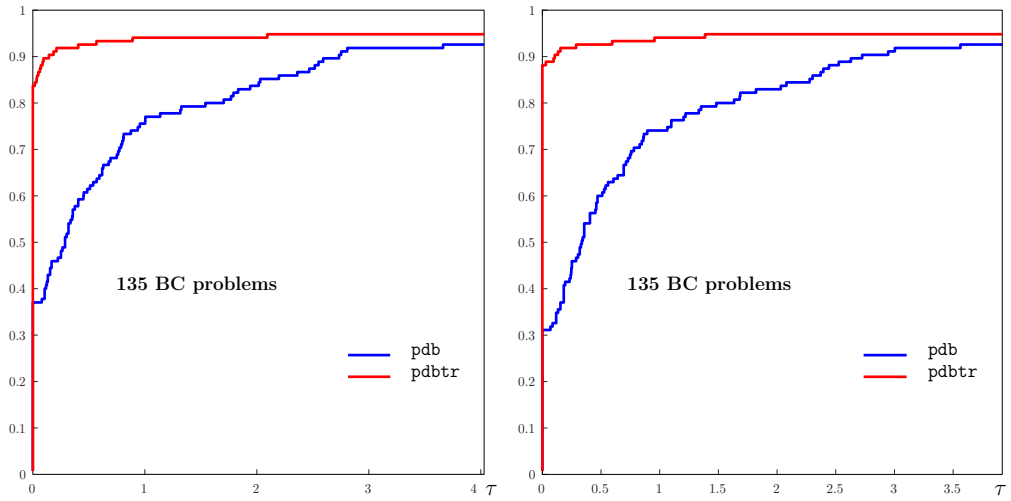

Figure 6.3. Performance profiles for the primal-dual interior algorithms `pdb` and `pdbtr` applied to 135 bound-constrained (BC) problems from the CUTEst test set.

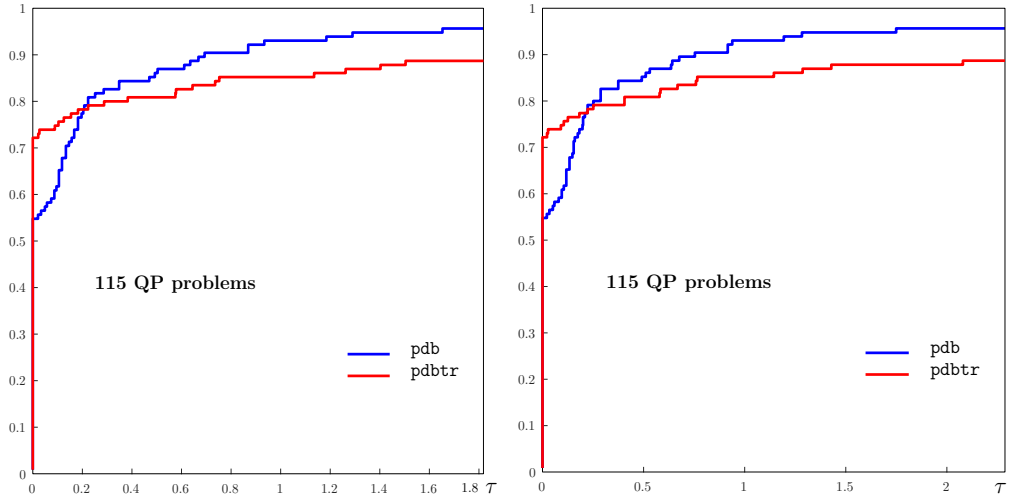

Figure 6.4. Performance profiles for the primal-dual interior algorithms `pdb` and `pdbtr` applied to 115 quadratic programs (QC) from the CUTEst test set.

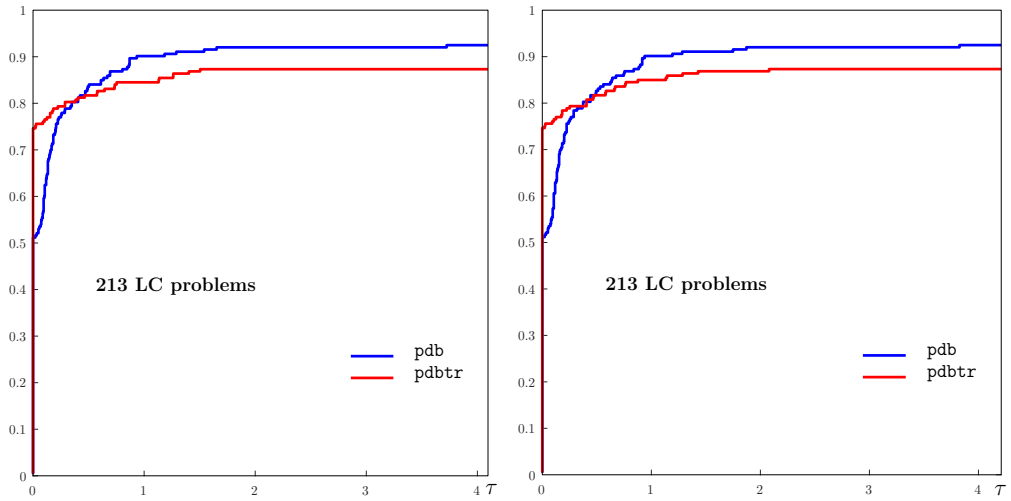

Figure 6.5. Performance profiles for the primal-dual interior algorithms `pdb` and `pdbtr` applied to 213 linearly-constrained (LC) problems from the CUTEst test set.

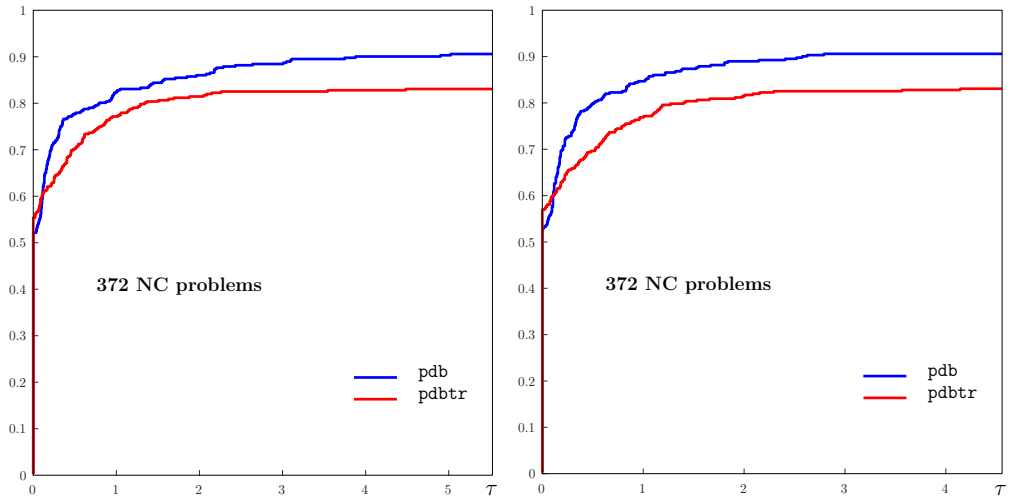

Figure 6.6. Performance profiles for the primal-dual interior algorithms `pdb` and `pdbtr` applied to 375 nonlinearly-constrained (NC) problems from the CUTEst test set.

Appendix A

Computation of Upper and Lower Bounds on Contraints and Variables

Generally, problems with mixed constraints can be written as

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & L_L c(x) \geq \ell^C,\ u^C \geq L_U c(x),\ L_X c(x) = h_X, \\ & E_L x \geq -\ell^X,\ u^X \geq -E_U x,\ E_X x = b_X, \end{aligned}$$

where L_L, L_U, E_L, E_U are matrices consisting of rows of the identity matrices. The reformulation and simplification of those equations will be complicated and they don't change the essence of the problem. Hence, for brevity, we only focus on the problems with upper and lower constraints:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & l^C \leq c(x) \leq u^C, l^X \leq x \leq u^X. \end{aligned}$$

We can introduce slack variables to rewrite the problem as follows:

$$\begin{aligned} \underset{x\in\mathbb{R}^n, l^X, u^X\in\mathbb{R}^n, l^C, u^C\in\mathbb{R}^m}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) - l^C + s \geq 0, u^C - c(x) + s \geq 0, s = 0, \\ & x - l^X + t \geq 0, u^X - x + t \geq 0, t = 0. \end{aligned}$$

The computations in the rest of the appendix involve long equations and big matrices, and hence will be done on landscape pages.

The merit function is

$$
\begin{aligned}
M(x,s,t,w,v,y_1,y_2,z_1,z_2;\mu^P,\mu^B) = f(x) - w^E s + \frac{1}{2\mu^P}\|s\|^2 + \frac{1}{2\mu^P}\|s+\mu^P(w-w^E)\|^2 \\
- v^E t + \frac{1}{2\mu^P}\|t\|^2 + \frac{1}{2\mu^P}\|t+\mu^P(v-v^E)\|^2 \\
- \sum_{j=1}^{m}\left\{\mu^B[y_1^E]_j \ln\Big([c(x)-l^C+s+\mu^B e]_j\Big) + \mu^B[y_1^E]_j \ln\Big([c(x)-l^C+s+\mu^B e]_j\cdot[y_1]_j\Big) - [c(x)-l^C+s+\mu^B e]_j\cdot[y_1]_j\right\} \\
- \sum_{j=1}^{m}\left\{\mu^B[y_2^E]_j \ln\Big([u^C-c(x)+s+\mu^B e]_j\Big) + \mu^B[y_2^E]_j \ln\Big([u^C-c(x)+s+\mu^B e]_j\cdot[y_2]_j\Big) - [u^C-c(x)+s+\mu^B e]_j\cdot[y_2]_j\right\} \\
- \sum_{i=1}^{n}\left\{\mu^B[z_1^E]_i \ln\Big([x-l^X+t+\mu^B e]_i\Big)\right\} + \mu^B[z_1^E]_i \ln\Big([x-l^X+t+\mu^B e]_i\cdot[z_1]_i\Big) - [x-l^X+t+\mu^B e]_i\cdot[z_1]_i\Big\} \\
- \sum_{i=1}^{n}\left\{\mu^B[z_2^E]_i \ln\Big([u^X-x+t+\mu^B e]_i\Big)\right\} + \mu^B[z_2^E]_i \ln\Big([u^X-x+t+\mu^B e]_i\cdot[z_2]_i\Big) - [u^X-x+t+\mu^B e]_i\cdot[z_2]_i\Big\}.
\end{aligned}
$$

As we did before, we introduce notations and approximations below:

$$W = \mathrm{diag}(w_i), \quad W^E = \mathrm{diag}(w_i^E), \quad ,Y_1^E = \mathrm{diag}(y_1^E), \quad Y_2^E = \mathrm{diag}(y_2^E), \quad Y_1 = \mathrm{diag}([y_1]_j), \quad Y_2 = \mathrm{diag}([y_2]_j).$$

$$C = \mathrm{diag}\Big(c_i(x)\Big), \quad X_1 = \mathrm{diag}([x-l^X+t+\mu^B e]_i), \quad X_2 = \mathrm{diag}([u^X-x+t+\mu^B e]_i), \quad D_{P_1} = \mu^P I_m, \quad D_{P_2} = \mu^P I_n.$$

$$S = \mathrm{diag}(s_i), \quad C_1 = \mathrm{diag}([c(x)-l^C+s+\mu^B e]_j), \quad C_2 = \mathrm{diag}([u^C-c(x)+s+\mu^B e]_j), \quad Y_1 = \mathrm{diag}([y_1]_j), \quad Y_2 = \mathrm{diag}([y_2]_j).$$

$$D_B = (C - \mathcal{L}^C + S + \mu^B I) Y_1^{-1}, \quad D_C = (U^C - C + S + \mu^B I) Y_2^{-1},$$

$$D_Q = \mu^B (C - \mathcal{L}^C + S + \mu^B I)^{-2} Y_1^E, \quad D_R = \mu^B (U^C - C + S + \mu^B I)^{-2} Y_2^E,$$

$$\pi^W = \pi^W(x,s) = w^E - \frac{1}{\mu^P} \cdot s, \quad \pi^V = \pi^V(x,t) = v^E - \frac{1}{\mu^P} \cdot t,$$

$$\pi^{Y_1} = \pi^{Y_1}(s) = \mu^B (C - \mathcal{L}^C + S + \mu^B I)^{-1} y_1^E, \quad \pi^{Y_2} = \pi^{Y_2}(s) = \mu^B (U^C - C + S + \mu^B I)^{-1} y_2^E,$$

$$\pi^{Z_1} = \mu^B (X - \mathcal{L}^X + T + \mu^B I_n)^{-1} Z_1^E, \quad \pi^{Z_2} = \mu^B (U^X - X + T + \mu^B I_n)^{-1} Z_2^E,$$

$$D_E = (X - \mathcal{L}^X + T + \mu^B I_n) Z_1^{-1}, \quad D_F = (U^X - X + T + \mu^B I_n) Z_2^{-1},$$

$$D_M = \mu^B (X - \mathcal{L}^X + T + \mu^B I_n)^{-2} Z_1^E, \quad D_N = \mu^B (U^X - X + T + \mu^B I_n)^{-2} Z_2^E,$$

with the approximation by substituting π^W with w, and π^{Y_1}, π^{Y_2} with y_1, y_2, and hence we get

$$D_Q = C_1^{-1} \cdot \mu^B C_1^{-1} Y_1^E = C_1^{-1} \Pi^{Y_1}$$

$$\approx C_1^{-1} Y_1 = D_B^{-1}, \qquad D_R = C_2^{-1} \cdot \mu^B C_2^{-1} Y_2^E = C_2^{-1} \Pi^{Y_2}$$

$$\approx C_2^{-1} Y_2 = D_C^{-1},$$

$$\mu^B Y_1^{-2} Y_1^E = D_B Y_1^{-1} \Pi^{Y_1} \approx D_B, \qquad \mu^B Y_2^{-2} Y_2^E = D_C Y_2^{-1} \Pi^{Y_2} \approx D_C,$$

$$D_M = X_1^{-1} \mu^B X_1^{-1} X_1^E = X_1^{-1} \Pi^{Z_1} \approx X_1^{-1} Z_1 = D_E^{-1}, \qquad D_N = X_2^{-1} \mu^B X_2^{-1} X_2^E = X_2^{-1} \Pi^{Z_2} \approx X_2^{-1} Z_2 = D_F^{-1},$$

$$\mu^B Z_1^{-2} Z_1^E = D_E Z_1^{-1} \Pi^{Z_1} \approx D_E, \qquad \mu^B Z_2^{-2} Z_2^E = D_F Z_2^{-1} \Pi^{Z_2} \approx D_F.$$

This gives us the fact that the perturbed Newton's equation coincide with the path-following equation, which is

$$\nabla F = \begin{pmatrix} H(x,y_1-y_2) & 0 & 0 & 0 & 0 & -J^T & J^T & -I_n & I_n \\ 0 & 0 & 0 & I_m & 0 & I_m & I_m & 0 & 0 \\ 0 & 0 & 0 & 0 & I_n & 0 & 0 & I_n & I_n \\ 0 & I_m & 0 & D_P & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_n & 0 & D_P & 0 & 0 & 0 & 0 \\ Y_1 J & Y_1 & 0 & 0 & 0 & C_1 & 0 & 0 & 0 \\ -Y_2 J & Y_2 & 0 & 0 & 0 & 0 & C_2 & 0 & 0 \\ Z_1 & 0 & Z_1 & 0 & 0 & 0 & 0 & X_1 & 0 \\ -Z_2 & 0 & Z_2 & 0 & 0 & 0 & 0 & 0 & X_2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta t \\ \Delta w \\ \Delta v \\ \Delta y_1 \\ \Delta y_2 \\ z_1 \\ z_2 \end{pmatrix} = - \begin{pmatrix} g - J^T y_1 + J^T y_2 - z_1 + z_2 \\ y_1 + y_2 + w \\ z_1 + z_2 + v \\ s + \mu^P (w - w^E) \\ t + \mu^P (v - v^E) \\ y_1 \cdot \left(c(x) - l^C + s \right) + \mu^B (y_1 - y_1^E) \\ y_2 \cdot \left(u^C - c(x) + s \right) + \mu^B (y_2 - y_2^E) \\ z_1 \cdot (x - l^X + t) + \mu^B (z_1 - z_1^E) \\ z_2 \cdot (u^X - x + t) + \mu^B (z_2 - z_2^E) \end{pmatrix}.$$

If we further introduce

$$\alpha = \Delta y_1 - \Delta y_2,$$

$$\beta = z_1 - z_2,$$

$$b = -D_B \pi^{Y_1} + D_{P_1} \pi^W + D_{P_1}(2y_1 + y_2),$$

$$c = -D_C \pi^{Y_2} - D_B \pi^{Y_1} + 2D_{P_1} \pi^W + 3D_{P_1}(y_1 + y_2),$$

$$e = -D_E \pi^{Z_1} + D_{P_2} \pi^V + D_{P_2}(2z_1 + z_2),$$

$$f = -D_F \pi^{Z_2} - D_E \pi^{Z_1} + 2D_{P_2} \pi^V + 3D_{P_2}(z_1 + z_2).$$

and

$$\Lambda = D_B + D_{P_1} - (D_B + D_C + 4D_{P_1})^{-1}(D_B + 2D_{P_1})^2,$$

$$\Theta = D_E + D_{P_2} - (D_E + D_F + 4D_{P_2})^{-1}(D_E + 2D_{P_2})^2,$$

$$\mu = b - (D_B + 2D_{P_1})(D_B + D_C + 4D_{P_1})^{-1} e,$$

$$\nu = e - (D_E + 2D_{P_2})(D_E + D_F + 4D_{P_2})^{-1} f,$$

we can have

$$\begin{pmatrix} H + \Theta^{-1} & -J^T \\ J & \Lambda \end{pmatrix} \begin{pmatrix} \Delta x \\ \alpha \end{pmatrix} = - \begin{pmatrix} g - J^T y_1 + J^T y_2 - z_1 + z_2 + \Theta^{-1} \nu \\ \mu \end{pmatrix},$$

where

$$\begin{aligned}
\beta &= -\Theta^{-1}(\Delta x - \nu),\\
(D_E + D_F + 4D_{P_2})z_2 &= -f - (D_E + 2D_{P_2})\beta,\\
(D_B + D_C + 4D_{P_1})\Delta y_2 &= -c - (D_B + 2D_{P_1})\alpha,\\
\Delta s &= \mu^P \pi^V + \mu^P(z_1 + z_2) + \mu^P(\Delta y_1 + \Delta y_2),\\
\Delta t &= \mu^P \pi^V + \mu^P(z_1 + z_2) + \mu^P(z_1 + z_2),\\
\Delta w &= w + y_1 + y_2 + \Delta y_1 + \Delta y_2,\\
\Delta v &= v + z_1 + z_2 + z_1 + z_2.
\end{aligned}$$

The dimension of the matrix equation has been siginificantly reduced, where the coefficient matrix is just a 2 by 2 block matrix. For the trust region problem, we can define

$$\bar{g} = \begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix}, \quad \bar{H} = \begin{pmatrix} H & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \bar{J} = \begin{pmatrix} 0 & I_m & 0 \\ 0 & 0 & I_n \\ J & I_m & 0 \\ -J & I_m & 0 \\ I_n & 0 & I_n \\ -I_n & 0 & I_n \end{pmatrix}, \quad \bar{D} = \begin{pmatrix} D_{P_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & D_{P_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & D_B & 0 & 0 & 0 \\ 0 & 0 & 0 & D_C & 0 & 0 \\ 0 & 0 & 0 & 0 & D_E & 0 \\ 0 & 0 & 0 & 0 & 0 & D_F \end{pmatrix}.$$

Then the equations can be written as

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

where

$$\bar{x} = \begin{pmatrix} x \\ s \\ t \end{pmatrix}, \quad \bar{y} = \begin{pmatrix} w \\ v \\ y_1 \\ y_2 \\ z_1 \\ z_2 \end{pmatrix}, \quad \bar{\pi} = \begin{pmatrix} \pi^W \\ \pi^V \\ \pi^{Y_1} \\ \pi^{Y_2} \\ \pi^{Z_1} \\ \pi^{Z_2} \end{pmatrix}.$$

If we choose $\bar{T}_x = I, \bar{T}_y = \bar{D}$, it can be written in the form

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

where $\bar{\sigma} = (1+\sigma)/(1+2\sigma)$, $\Delta\hat{y} = (1+2\sigma)\Delta\bar{y}$.

Define

$$\alpha = \Delta\tilde{y}_1 - \Delta\tilde{y}_2, \quad \beta = \Delta\tilde{z}_1 - \Delta\tilde{z}_2,$$
$$b = D_B(y_1 - \pi^{Y_1}) - \sigma^{-1}(w + y_1 + y_2),$$
$$c = D_C(y_2 - \pi^{Y_2}) + D_B(y_1 - \pi^{Y_1}) - 2\sigma^{-1}(w + y_1 + y_2),$$
$$e = D_E(z_1 - \pi^{Z_1}) - \sigma^{-1}(v + z_1 + z_2),$$
$$f = D_F(z_2 - \pi^{Z_2}) + D_E(z_1 - \pi^{Z_1}) - 2\sigma^{-1}(v + z_1 + z_2),$$

and

$$\Lambda = \bar{\sigma}D_B + 2\sigma^{-1}I_m - [\bar{\sigma}D_B + \bar{\sigma}D_C + 4\sigma^{-1}I_m]^{-1}(\bar{\sigma}D_B + 2\sigma^{-1}I_m)^2,$$
$$\Theta = \bar{\sigma}D_E + 2\sigma^{-1}I_n - [\bar{\sigma}D_E + \bar{\sigma}D_F + 4\sigma^{-1}I_n]^{-1}(\bar{\sigma}D_E + 2\sigma^{-1}I_n)^2,$$
$$\mu = b - [\bar{\sigma}D_B + \bar{\sigma}D_C + 4\sigma^{-1}I_m]^{-1}(\bar{\sigma}D_B + 2\sigma^{-1}I_m)c,$$
$$\nu = e - [\bar{\sigma}D_E + \bar{\sigma}D_F + 4\sigma^{-1}I_n]^{-1}(\bar{\sigma}D_E + 2\sigma^{-1}I_n)f.$$

We can again simplify the matrix equation as

$$\begin{pmatrix} H + \Theta^{-1} & -J^T \\ J & \Lambda \end{pmatrix} \begin{pmatrix} \Delta x \\ \alpha \end{pmatrix} = - \begin{pmatrix} g - J^T y_1 + J^T y_2 - z_1 + z_2 + \Theta^{-1}\nu \\ \mu \end{pmatrix},$$

which again only involves a 2 by 2 block matrix.

Bibliography

[1] Roberto Andreani, José Mario Martínez, and B. F. Svaiter. A new sequential optimality condition for constrained optimization and algorithmic consequences. *SIAM J. Optim.*, 20(6):3533–3554, 2010.

[2] I. Bongartz, A. R. Conn, N. I. M. Gould, and Philippe L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Trans. Math. Software*, 21(1):123–160, 1995.

[3] Marc G. Breitfeld and David F. Shanno. Computational experience with penalty-barrier methods for nonlinear programming. *Ann. Oper. Res.*, 62:439–463, 1996.

[4] 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.

[5] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Math. Comput.*, 66(217):261–288, 1997.

[6] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-Region Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[7] Gerard Debreu. Definite and semidefinite quadratic forms. *Econometrica*, 20:295–300, 1952.

[8] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002.

[9] Anders Forsgren. Inertia-controlling factorizations for optimization algorithms. *Appl. Numer. Math.*, 43:91–107, 2002.

[10] Anders Forsgren and Philip E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. Optim.*, 8:1132–1152, 1998.

[11] E. Michael Gertz. *Combination Trust-Region Line-Search Methods for Unconstrained Optimization*. PhD thesis, Department of Mathematics, University of California, San Diego,

1999.

[12] E. Michael Gertz and Philip E. Gill. A primal-dual trust-region algorithm for nonlinear programming. *Math. Program., Ser. B*, 100:49–94, 2004.

[13] Philip E. Gill, Vyacheslav Kungurtsev, and Daniel P. Robinson. A shifted primal-dual penalty-barrier method for nonlinear optimization. *SIAM J. Optim.*, 30(2):1067–1093, 2020.

[14] Philip E. Gill and Daniel P. Robinson. A primal-dual augmented Lagrangian. Numerical Analysis Report 08-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2008.

[15] Philip E. Gill and Daniel P. Robinson. A globally convergent stabilized SQP method. *SIAM J. Optim.*, 23(4):1983–2010, 2013.

[16] Philip E. Gill and Elizabeth Wong. Methods for convex and general quadratic programming. *Math. Program. Comput.*, 7:71–112, 2015.

[17] Philip E. Gill and Margaret H. Wright. *Computational Optimization: Nonlinear Programming*. Cambridge University Press, New York, NY, USA, 2020. To be published in 2023.

[18] Donald Goldfarb, Roman A. Polyak, Katya Scheinberg, and I. Yuzefovich. A modified barrier-augmented Lagrangian method for constrained minimization. *Comput. Optim. Appl.*, 14(1):55–74, 1999.

[19] Nicholas I. M. Gould. On modified factorizations for large-scale linearly constrained optimization. *SIAM J. Optim.*, 9:1041–1063, 1999.

[20] Nicholas I. M. Gould, D. Orban, and Philippe L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Software*, 29(4):373–394, 2003.

[21] William W. Hager. Condition estimates. *SIAM J. Sci. Statist. Comput.*, 5(2):311–316, 1984.

[22] M. D. Hebden. An algorithm for minimization using exact second derivatives. Technical Report T.P. 515, Atomic Energy Research Establishment, Harwell, England, 1973.

[23] Nicholas J. Higham. FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation. *ACM Trans. Math. Software*, 14:381–396, 1988.

[24] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*. Lecture

Notes in Econom. Math. Syst. 187. Springer-Verlag, Berlin, 1981.

[25] Jorge J. Moré and Danny C. Sorensen. Computing a trust region step. *SIAM J. Sci. and Statist. Comput.*, 4:553–572, 1983.

[26] Stephen G. Nash, Roman Polyak, and Ariela 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.

[27] Jorge Nocedal and Ya-Xiang 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.

[28] Roman A. Polyak. Modified barrier functions (theory and methods). *Math. Program.*, 54(2, Ser. A):177–222, 1992.

[29] Michael J. D. Powell. On the global convergence of trust region algorithms for unconstrained minimization. *Math. Programming*, 29(3):297–303, 1984.

[30] C. H. Reinsch. Smoothing by spline functions II. *Numer. Math.*, 16:451–454, 1971.

[31] Andreas Wächter and Lorenz T. Biegler. Line search filter methods for nonlinear programming: motivation and global convergence. *SIAM J. Optim.*, 16(1):1–31 (electronic), 2005.

[32] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1, Ser. A):25–57, 2006.