Computing the Local Minimizers of a Large and Sparse Trust-Region Subproblem

Charles Fortin

Department of Mathematics and Statistics,
McGill University, Montréal
Québec, Canada

October, 2004

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Your file Votre référence
ISBN: 0-494-12841-0
Our file Notre référence
ISBN: 0-494-12841-0

Abstract

We present new algorithms for computing local minimizers of the trust-region subproblem (TRS). This problem consists in minimizing a quadratic function subject to a ball constraint. In particular, this problem appears as a subproblem in trust-region methods for constrained and unconstrained optimization. First, by modeling the TRS with a new semidefinite program, different than the standard semidefinite relaxation, we derive an algorithm, similar in structure to the Rendl-Wolkowicz Algorithm, which implicitly solves the semidefinite program by maximizing a single variable concave function over a closed interval. Second, we extend the theory needed for this algorithm and the Rendl-Wolkowicz Algorithm to derive two algorithms for computing a local-nonglobal minimizer of the TRS. These algorithms are based upon finding a root of a single variable convex function with the secant method. In all our algorithms, we compute at most the two smallest eigenvalues of a parameterized matrix. This can be done using an ARPACK subroutine which only requires matrix-vector multiplications. Hence, we are able to exploit the possible sparsity of the Hessian matrix of the quadratic objective, making the algorithms suitable for large problems. Computationally, the algorithms for finding a local-nonglobal minimizer are more competitive than the previous approach based on computing a matrix factorization at each iteration.

Résumé

Nous présentons de nouveaux algorithmes pour calculer les minimiseurs locaux d'un sous-problème de région de confiance (SRC). Ce problème consiste à minimiser une fonction quadratique à l'intérieur d'une boule. En particulier, ce problème est un sous-problème pour les méthodes de région de confiance en optimisation avec et sans contraintes. Premièrement, en modélisant le SRC par le biais d'un nouveau programme semi-defini, différent de la relaxation semi-definie usuelle, nous proposons un algorithme, similaire en structure à l'algorithme de Rendl-Wolkowicz, qui résout implicitement le programme semi-defini en maximisant une fonction concave d'une variable sur un intervalle fermé. Deuxièmement, nous élargissons la théorie nécessaire à cet algorithme et l'algorithme de Rendl-Wolkowicz pour proposer deux algorithmes visant à calculer un minimiseur local non-global du SRC. Ces algorithmes trouvent une racine d'une fonction convexe d'une variable par la méthode de la sécante. Dans tous nos algorithmes, nous calculons au plus les deux plus petites valeurs propres d'une matrice parametrée. Ceci peut être calculé grâce à une sous-routine d'ARPACK qui ne nécessite que des produits matrice-vecteur. Donc, nous sommes capable de profiter de la faible densité potentielle de la matrice hessienne de la fonction objective quadratique, rendant les algorithmes aptes à résoudre des problèmes de grande dimension. Pour les calculs sur ordinateur, les algorithmes permettant de trouver le minimiseur local non-global sont plus performants que l'approche précédente utilisant une factorisation matricielle à chaque iteration.

Acknowledgments

J'aimerais remercier toutes les personnes, ami(e)s et famille, qui ont cru en moi et cette thèse et qui ont su m'épauler et m'encourager. Merci à mon superviseur Jean-Louis Goffin et à Henry Wolkowicz pour avoir stimulé ma réflexion. Enfin, merci au Fond de recherche sur la nature et technologies du Québec, à l'Institut des sciences mathématiques et à Jean-Louis Goffin pour leurs aides financières.

Table of Contents

Introduction

Consider the following unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where f is a smooth function. When computing a pure Newton step from a given estimate x_0 of the solution of problem (1), one needs to solve the optimization problem

$$\min_{d \in \mathbb{R}^n} \nabla f(x_0)^T d + \frac{1}{2} d^T \nabla^2 f(x_0) d. \tag{2}$$

The standard iteration is equivalent to finding the optimal d where the gradient of the quadratic model of f at x_0 is zero. It is possible however that problem (2) is unbounded or that the quadratic model is a poor representation of f at the minimizer of problem (2). These situations lead to a lack of global convergence for Newton's Method.

A possible globalization scheme is to include a ball constraint, equivalently a *trust-region*, to problem (2), i.e. at each iteration, pick a well-chosen radius Δ, and solve

$$\begin{aligned} \min \quad & \nabla f(x_0)^T d + \tfrac{1}{2} d^T \nabla^2 f(x_0) d \\ \text{s.t.} \quad & \|d\| \leq \Delta. \end{aligned} \tag{3}$$

Then a step $x_1 = x_0 + d$ is taken under the condition that this yields an appropriate decrease in the function f. The size of the trust-region is ajusted accordingly. Under appropriate conditions, this method, called a *trust-region method* will converge to a solution satisfying first and second order optimality conditions for problem (1) [8].

Consider now the problem with equality constraints

$$\begin{aligned} \min \quad & g(x) \\ \text{s.t.} \quad & c(x) = 0, \end{aligned} \tag{4}$$

where $c \in \mathbb{R}^m$ and c and g are smooth functions. Let $\mathcal{L}(x, \lambda) = g(x) - \lambda^T c(x)$ be the Lagrangian function for this problem. Given again an estimate x_0 of the solution of problem (4) and an estimate λ_0 of a Lagrange multiplier (assuming a constraint qualification insures such a multiplier exists), one may apply Newton's Method to the first-order optimality conditions in order to compute the next iterate. If we assume full row rank of the constraint Jacobian $\nabla c(x_0)$, this can be showed equivalent to solving the quadratic program

$$\begin{aligned} \min \quad & \nabla g(x_0) + \tfrac{1}{2} d^T \nabla^2_{xx} \mathcal{L}(x_0, \lambda_0) d \\ \text{s.t.} \quad & \nabla c(x_0) d + c(x_0) = 0. \end{aligned} \tag{5}$$

However, it is possible that linearizing the constraints makes problem (5) infeasible or that the quadratic program is a poor model of problem (4) at the minimizer of problem (5). One possible "fix" is not to try to satisfy the equality constraint at each step, but to improve the feasibility of the constraint, and to include a trust-region. Thus, given well-chosen positive scalars π and Δ, we solve the quadratically constrained quadratic problem

$$\begin{aligned} \min \quad & \nabla g(x_0) + \tfrac{1}{2} d^T \nabla^2_{xx} \mathcal{L}(x_0, \lambda_0) d \\ \text{s.t.} \quad & \|\nabla c(x_0) d + c(x_0)\|^2 \leq \pi^2, \\ & \|d\|^2 \leq \Delta^2. \end{aligned} \tag{6}$$

This problem is often referred to as the *two trust-region subproblem* and the idea presented is the basis of trust-region sequential quadratic programming methods.

This thesis proposes different algorithms for computing the local minimizers of the *trust-region subproblem* (TRS)

$$\text{(TRS)} \qquad \begin{aligned} \min_x \quad & x^T A x - 2a^T x \\ \text{s.t.} \quad & \|x\| \leq \Delta. \end{aligned} \tag{7}$$

Here, A is an $n \times n$ symmetric (possibly indefinite) matrix, a is an n-vector, x is the n-vector of unknowns and the ball radius Δ is a positive scalar. All matrix and vector entries are real. We focus on cases where A is a large and sparse matrix. In order to exploit the sparsity, our algorithms will require only matrix-vector products $Aw \leftarrow w$. In this sense, the algorithms are *matrix-free.*

Solutions to the TRS will be referred to as *global minimizers.* However, this problem may possess a unique local minimizer which is not a global minimizer. Such feasible solution will be referred to as a *local-nonglobal minimizer.* Clearly, if we can compute a global minimizer, then we can solve problem (3). Moreover, if both constraints of problem (6) are not active at an optimal solution for this problem, then this optimal solution is a local minimizer (global or local-nonglobal) of a TRS such as (7). Problems (3) and (6) thus motivated our interest in the local-minimizer of the TRS. In Chapter 1, we mention other contexts where this problem also appears.

The algorithms we present are motivated by a simple geometric interpretation for the TRS, in the case where solutions are located on the boundary of the trust-region. We show that each local minimizer located on the boundary of the trust-region lies also on the boundary of an ellipsoid locally contained, near the local minimizer, in the trust-region. This ellipsoid is an element of a family of ellipsoids having for boundary the level curves of a convex quadratic function, obtained by properly shifting the eigenvalues of the matrix A in the quadratic objective function of TRS.

When computing a global minimizer for TRS, the two main difficulties are exploiting the sparsity of the matrix A and handling the so-called *hard case.* The algorithm we propose builds on the method of Rendl and Wolkowicz [10, 39], which has overcome these two major concerns. These authors have shown that TRS can be equivalently solved by maximizing a concave function of one variable $k(t)$, where

$$k(t) = (\Delta^2 + 1)\lambda_1(D(t)) - t, \qquad D(t) = \begin{pmatrix} t & -a^T \\ -a & A \end{pmatrix}$$

and $t \in \mathbb{R}$. Sparsity of the matrix A may be exploited, because the main efforts in their algorithm lie in computing $\lambda_1(D(t))$, the smallest eigenvalue of $D(t)$, and a corresponding eigenvector using the Lanczos algorithm, which requires only matrix-vector multiplications. The hard case is handled in two ways: first by taking steps to the boundary from feasible points of TRS in a similar way to the Moré and Sorensen Algorithm [32] and second by a *shift and deflate* procedure based on the result that every hard case TRS may be reduced to an equivalent easy case TRS [10].

Similarly, we show TRS can be equivalently solved by maximizing a concave function of one variable $f(\gamma)$, where

$$f(\gamma) := \gamma\lambda_1(B(\gamma)), \qquad B(\gamma) := \Delta^2 B - \frac{1}{1-\gamma} B^{-1/2} a a^T B^{-1/2}, \tag{8}$$

and $\gamma \in (-\infty, 1)$. Here $B = A - \bar{\lambda} I$ for a well chosen value of $\bar{\lambda}$ which makes B positive definite. Computing $\lambda_1(B(\gamma))$ may be obtained through an ARPACK subroutine which exploits sparsity of the matrix A. By mimicking the Rendl-Wolkowicz Algorithm, our algorithm takes steps to the boundary from feasible points of TRS and also applies the shift procedure developed in [10]. Thus, the algorithm proposed exploits the recent advances for taking advantage of the sparsity of the matrix A and handling hard case TRS. There is also one more link between our algorithm and the Rendl-Wolkowicz Algorithm. In the latter algorithm, maximizing k is showed to be equivalent to solving the dual of the semidefinite relaxation of TRS [39]. We show maximizing f is equivalent to solving a different semidefinite program that comes out of a result related to extremal ellipsoids.

An algorithm for computing a local-nonglobal minimizer was first proposed by Martínez [29]. In addition to his algorithm, he presents surprising results. In particular, there is not more than one local-nonglobal minimizer.

The two algorithms we propose for computing the local-nonglobal minimizers of TRS respectively build on the theory behind our algorithm for computing a global

minimizer and the Rendl-Wolkowicz Algorithm. We also exploit the results of Martínez [29]. Each algorithm share the same structure: we compute the largest root of a strictly convex function using the secant method. Evaluating the function requires finding the two smallest eigenvalues of the matrix $B(\gamma)$ or $D(t)$, depending on the algorithm. Again those are obtained through an ARPACK subroutine which exploits sparsity of the matrix A.

The thesis is organized as follow. In Chapter 1, we review some of the algorithms which have appeared in the literature to compute local minimizers of the TRS. We also motivate the interest in the problem with some applications.

In Chapter 2, we derive a new algorithm for computing a global minimizer of the TRS. We initially review the optimality conditions and reformulate the TRS using maximal ellipsoids. In particular, this leads to a semidefinite program, different from the usual semidefinite relaxation associated with TRS, which is implicitly solved with our algorithm. A convergence result for a simplified version of our algorithm is presented. Numerical results appear at the end of the chapter, where we compare our algorithm with other recent algorithms and where we briefly show how TRS algorithms may be used as subroutines within trust-region methods for solving unconstrained and constrained optimization problems.

In Chapter 3, we start by reviewing the results obtained by Martínez [29] on the local non-global minimizer. We then present successively the two new algorithms for computing this minimizer. We end the chapter by comparing all three algorithms through some numerical experiments.

0.1 Notation

We will use the following standard notation throughout the thesis. All norms are two-norms. The identity matrix is I. The space of $n \times m$ real matrices is denoted by

$\mathbb{M}^{n,m}$. For $M \in \mathbb{M}^{n,m}$, we denote its transpose by M^T and its null space by $\mathcal{N}(M)$. If $n = m$, we denote its inverse by M^{-1}, its Moore-Penrose generalized inverse by $M^\dagger$, its determinant by $\det(M)$ and its trace by $\mathrm{tr}\,(M)$. Given a symmetric matrix $S \in \mathbb{M}^{n,n}$, $\lambda_j(S)$ denotes the j-th smallest eigenvalue of S, where $1 \leq j \leq n$. Thus

$$\lambda_1(S) \leq \lambda_2(S) \leq \ldots \leq \lambda_n(S).$$

If S is positive semidefinite (definite), we use $S \succeq 0$ ($S \succ 0$). We denote its square root by $S^{1/2}$ and the inverse of the latter matrix by $S^{-1/2}$. If $\mathcal{Q}$ is a set of vectors in $\mathbb{R}^n$, $x \in \mathbb{R}^n$ and $x^T q = 0$ for all $q \in \mathcal{Q}$, we write $x \perp \mathcal{Q}$ and, $x \not\perp \mathcal{Q}$ otherwise. When we write $x \searrow a$, we mean x converges to a and $x > a$. Similarly, $x \nearrow a$ means x converges to a and $x < a$. For a function $f : \mathbb{R} \mapsto \mathbb{R}$, we define

$$f(a^+) := \lim_{x \searrow a} f(x), \qquad f(a^-) := \lim_{x \nearrow a} f(x).$$

Chapter 1

Literature Review

1.1 Global Minimizers

Finding a minimizer of a quadratic function subject to a ball constraint is an important problem in optimization and linear algebra. Some applications of this problem are the computation of a new iterate in trust-region methods for unconstrained optimization [36, 37, 42], the computation of quadratically constrained least squares [11, 16], the regularization of discrete forms of ill-posed problems [41], and ridge regressions [22]. Because it is the main subroutine in trust-region methods for unconstrained optimization, the problem is called the trust-region subproblem. Recent reviews of this problems are partially included in the papers of Yuan [55] and Fortin and Wolkowicz [10]. For an introduction to the trust-region subproblem and trust-region methods in general, the book by Nocedal and Wright [34] on numerical optimization is a well-written reference intended for the graduate level. For a deeper treatment of trust-regions methods, a monumental book was written by Conn, Gould and Toint [8], where many pages are devoted to the trust-region subproblem.

The idea of minimizing a function by constructing a quadratic model and forcing the minimization to take place in a neighborhood represented by an Euclidean ball

can be traced back as far as 1944. In a paper by Levenberg [26], this concept is introduced for solving least square problems encountered in curve fitting. A similar idea is later presented by Marquardt [28] where using a Euclidean norm constraint explicitly first appears. Due to the origin of the problems considered, these last authors dealt only with convex quadratics. It is in 1966 that Goldfeld, Quandt and Trotter [14] considered minimizing a non-convex quadratic objective subject to a ball constraint.

For a computationally efficient method that could handle the so-called *hard case*, one needed to wait until the early 1980's for the work of Gay [12], where necessary and sufficient optimality conditions first appeared, and Moré and Sorensen [32]. The algorithm of Moré and Sorensen uses Newton's method for finding a root of the single-variable function

$$\frac{1}{\Delta} - \frac{1}{\|(A - \lambda I)^{-1} a\|},$$

which naturally arises from the optimality conditions. Cholesky factorizations are used to computed the Newton directions and steps to the boundary are taken to handle the hard case. The algorithm is well-suited for small size trust-region subproblems and particularly effective in the hard case. However, the use of Cholesky factorizations limits the size of the problems that can be considered in practical applications (though for some problems, a sparse Cholesky algorithm may work).

Later on, the focus was turned toward methods that could exploit the sparsity of the matrix A, which appears in the quadratic objective. Methods of choice were those that would require only matrix-vector products $w \leftarrow Aw$. Among the first algorithms of this new generation of algorithms are the conjugate gradient based algorithm of Steihaug [44] and Toint [49]. These algorithms do not compute a nearly optimal solution to the trust-region subproblem, but an approximate solution at least as good as the *Cauchy point* (the minimizer along the steepest descent direction) is obtained.

Since the late 1990's there has been a wide range of newly proposed algorithms.

The algorithm of Sorensen [42] has a superlinearly convergent two-point scheme in the easy case based upon a well-chosen interpolant of the function

$$\varphi(\lambda) := \|(A - \lambda I)^{-1} a\|^2.$$

In the hard case, a linearly convergent one-point scheme is used and steps to the boundary are taken from feasible points. Each iterate is obtained by computing the smallest eigenvalue and an associated eigenvector of the bordered matrix

$$D(t) = \begin{pmatrix} t & -a^T \\ -a & A \end{pmatrix}.$$

Sparsity may be exploited with the use of a Lanczos method for computing the desired eigenvalues.

This is not the only algorithm which recasts the trust-region subproblem in terms of a parametric eigenvalue problem. As mentioned in the introduction, the algorithm of Rendl and Wolkowicz [39] reformulates the trust-region subproblem as an unconstrained minimization of the single-variable function

$$k(t) = (\Delta^2 + 1)\lambda_1(D(t)) - t.$$

Here again, one needs at each iterate t to compute the smallest eigenvalue of the bordered matrix $D(t)$ and an associated eigenvector in order to evaluate the function $k(t)$ and its first derivative.

As an extension to the algorithm of Sorensen, the algorithm of Rojas, Santos and Sorensen [40] yields a superlinearly convergent two-point scheme based however on a different interpolant of $\varphi(\lambda)$. The method converges in both the easy and hard case. In particular, the hard case is handled by computing for each iterate two eigenvalues and associated eigenvectors of the bordered matrix $D(t)$ corresponding to the smallest eigenvalue and an eigenvalue close to the second smallest eigenvalue.

Another possibility for solving the trust-region subproblem is to restrict the variable x to lie in a specially constructed subspace, i.e. solve

$$\begin{aligned} \min \quad & q(x) \\ \text{s.t.} \quad & \|x\| \leq \Delta \\ & x \in S. \end{aligned}$$

In the algorithm of Gould, Lucidi, Roma and Toint [17], S is the Krylov subspace

$$S = \{a, Aa, A^2a, A^3a, \ldots, A^ka\},$$

where k is increased at each iteration. The Lanczos method is used to obtain an orthonormal basis of this subspace. At each iteration, the Moré-Sorensen Algorithm is used to solve a well-structured trust-region subproblem, where the matrix in the quadratic objective is tridiagonal. The Cholesky factorizations required may take advantage of this structure in order to exploit sparsity. However, this algorithm does not handle the hard case. Another choice of S is also suggested by Hager [19]. His algorithm keeps the size of S to a low dimension of four. At an iterate $\bar{x}$, the subspace S is the span of the vector $\bar{x}$, the gradient $A\bar{x} - a$, an approximate eigenvector for the smallest eigenvalue of A and the iterate obtained by applying one step of a sequential quadratic programming algorithm to the trust-region subproblem. The resulting method is proved to converge quadratically.

Finally, the algorithm of Tao and An [47, 48] exploits the fact that the quadratic objective function may be rewritten as the difference of convex functions, i.e.

$$q(x) = (\rho\|x\|^2 - 2a^Tx) - (x^T(\rho I - A)x),$$

where ρ is a fixed constant chosen so that $\rho I - A$ is positive definite. The resulting algorithm is simple, but we are only guaranteed that a limit point of the sequence of iterates is a local minimizer. Yet, the authors show how, from a local-nonglobal minimizer, one may move to a feasible point with a smaller objective value where the method may be restarted.

1.2 Local-Nonglobal Minimizer

The interest in the local-nonglobal minimizer of problem (7) is that it may be the global minimizer of the following problem:

$$\begin{aligned} \min_{x} \quad & x^T Ax - 2a^T x \\ \text{s.t.} \quad & h(x) \leq 0, \\ & \|x\| \leq \Delta, \end{aligned} \tag{1.1}$$

where $h : \mathbb{R}^n \to \mathbb{R}^m$. Indeed, if x^* is a global minimizer of the latter problem and $h(x^*) < 0$, then clearly x^* is a local-minimizer of problem (7). In this case, x^* is a local-nonglobal minimizer of problem (7) if all global minimizers of problem (7) do not satisfy the first constraint of problem (1.1). An important special case known as the *two trust-region subproblem* is when h is a convex quadratic [1, 21]. This problem appears while computing the Celis-Dennis-Tapia problem [6, 38, 54] in a sequential quadratic programming approach for solving nonlinear programs. It also appears as a subproblem in the numerical solution of parameter identification problems of the form

$$\begin{aligned} \min_{x} \quad & \|F(x) - y\|^2 \\ \text{s.t.} \quad & \|x\| \leq \Delta, \end{aligned}$$

see [20, 21]. For this subproblem, the constraints in (1.1) are two balls constraints. More generally, Martínez and Santos [30] described an algorithm for minimizing a differentiable function over an Euclidean ball, where minimizing a quadratic function over the intersection of two Euclidean balls also appears as a subproblem. Finally, extensions to general quadratic constraints (possibly indefinite) is discussed in [31, 35, 45].

It is important to mention, even in the case where h is a convex quadratic, that there are no known polynomial time algorithm for computing the global minimum of problem (1.1). It is not known either if the problem is NP-hard. If this happens to

be true, in general one may only expect approximate solutions. However, in special cases, it is possible to compute a solution as close as we want to the exact solution. For example, the paper of Zhang and Ye [53] combines the matrix decomposition result of Sturm and Zhang [46] and semidefinite relaxation to show that some cases of the two trust-region subproblem may be solved. They even propose an algorithm, for the two trust-region subproblem, which follows the path of solutions of a family of parameterized problems. However, no convergence result is proved for this algorithm, but it is illustrated on some examples.

There is very little written on computing local-nonglobal minimizers for the trust-region subproblem. In fact, the author of this thesis is only aware of the algorithm of Martínez [29]. The algorithm is based on finding the smallest root of the convex function $\varphi(\lambda) - \Delta^2$ on the interval $(\lambda_1(A), \lambda_2(A))$ where lies the associated Lagrange multiplier of the local-nonglobal minimizer. At each iterate, a full spectral decomposition of the matrix A or a matrix factorization of a related matrix is required to evaluate the function $\varphi(\lambda)$ and compute its first two derivatives; the algorithm is not designed to exploit sparsity of the matrix A.

A last refence on local-nonglobal minimizers is the paper of Lucidi, Palagi and Roma [27]. The paper shows in particular that strict complementarity holds at a local-nonglobal minimizer.

Chapter 2

Global Minimizers

2.1 Optimality Conditions and Assumptions

Recall the trust region subproblem is

$$\text{(TRS)} \qquad \begin{aligned} \min_x \quad & q(x) := x^T A x - 2a^T x \\ \text{s.t.} \quad & \|x\| \leq \Delta, \end{aligned} \tag{2.1}$$

and consider the equality constrained trust-region subproblem

$$\text{(TRS=)} \qquad \begin{aligned} \min_x \quad & q(x) = x^T A x - 2a^T x \\ \text{s.t.} \quad & \|x\| = \Delta. \end{aligned} \tag{2.2}$$

Without loss of generality, we assume in this thesis, unless mentioned otherwise, $\Delta = 1$, since we may scale the matrix A, the vector a and the vector x, respectively by Δ^2, Δ and $1/\Delta$ so that we end up minimizing over the unit ball. This is done to simplify the notation.

Unless

$$A \succ 0 \qquad \text{and} \qquad \|A^{-1}a\| < 1, \tag{2.3}$$

an optimal solution to problem (2.2) will be also optimal for (2.1) [32]. The proof of this statement relies on the fact that a solution to the equation $Ax - a = 0$ has

the same objective value as $x + z$, if $z \in \mathcal{N}(A)$. Thus, if the conditions (2.3) are not satisfied and x^* solves (2.2), then x^* solves (2.1). We shall focus our attention on problem (2.2) and return to problem (2.1) later. We will actually consider a slight modified problem. Since the feasibility constraint of problem (2.2) allows us to write $q(x) = x^T(A - \lambda_1(A)I)x - 2a^Tx + \lambda_1(A)$, if we replace the matrix A by the matrix $A_r := A - \lambda_1(A)I$ in (2.2), the optimal solutions x^* are unchanged. We thus reset A to the latter matrix (note $\lambda_1(A_r) = 0$) and consider the problem

$$\begin{aligned} \min_x \quad & q(x) = x^T A_r x - 2a^T x \\ \text{s.t.} \quad & \|x\| = 1. \end{aligned} \tag{2.4}$$

We assume for now $a \neq 0$. This is justified since $a = 0$ makes (2.2) or (2.4) simple eigenvalue problems. In the context of trust region methods for unconstrained optimization, this assumption also makes sense, since from problem (3) we see $a = \nabla f(x_0)$, and these methods usually stop when $\nabla f(x_0)$ is close to zero.

It is well-known [13, 42] that x^* is a global minimizer to (2.4) if and only if

$$(A_r - \lambda^* I)x^* = a, \tag{2.5a}$$

$$A_r - \lambda^* I \succeq 0, \tag{2.5b}$$

$$\|x^*\| = 1. \tag{2.5c}$$

for some (unique) Lagrange multiplier $\lambda^* \in \mathbb{R}$. For x^* to be a minimizer of problem (2.1), we need to add the complementary slackness equation $\lambda^*(\|x^*\| - 1) = 0$ and the correct sign for the multiplier, i.e. $\lambda^* + \lambda_1(A) \leq 0$.

Following [39], we use throughout the paper the terminology, given in Table 2.1, to define three possible instances for (2.4). We use the term *hard case* when we are not in the easy case. Then we may either be in the hard case 1 or the hard case 2.

Equation (2.5) shows that, in the easy case, the hard case 1 and the hard case

Easy case	Hard case 1	Hard case 2
$a \not\perp \mathcal{N}(A_r)$	$a \perp \mathcal{N}(A_r)$	$a \perp \mathcal{N}(A_r)$
	and	and
(implies $\lambda^* < 0$)	$\lambda^* < 0$	$\lambda^* = 0$
		(i) $\|A_r^\dagger a\| = 1$
		(ii) $\|A_r^\dagger a\| < 1$

Table 2.1: The three different cases for the equality trust region subproblem (2.4). We include two sub-cases for the hard case 2.

2(i), $x^* = x(\lambda^*)$, where we define

$$x(\lambda) := (A_r - \lambda I)^\dagger a,$$

and where λ^* (uniquely) solves, for $\lambda \in (-\infty, 0]$, $\|x(\lambda)\| = 1$. The hard case 2(ii) is characterized by the fact that there exists $R < 1$ such that $\|x(\lambda)\| \leq R$, for all $\lambda \in (-\infty, 0]$. In this case, an optimum is given by

$$x^* = A_r^\dagger a + \alpha z, \tag{2.6}$$

where α is chosen to satisfy $\|x^*\| = 1$ and $z \in \mathcal{N}(A_r)$. In fact, the hard case 2(ii) is what is referred to as the *hard case* by Moré and Sorensen [32], because it is truly this case which forces the sophistication of their algorithm. However, even though our techniques to handle the hard case are also mostly designed to treat the hard case 2(ii), we prefer the notation of Table 2.1 introduced in [39].

2.2 Reformulating TRS Using Maximal Ellipsoids

Since $\lim_{\lambda\to-\infty} \|(A_r - \lambda I)^{-1}a\| = 0$, let $\bar{\lambda}$ be such that

$$B \succ 0, \tag{2.7a}$$

$$\|B^{-1}a\| \leq 1, \tag{2.7b}$$

where we define $B := A_r - \bar{\lambda}I$. For x feasible for (2.4) , $q(x) = x^TBx - 2a^Tx + \bar{\lambda}$, and, by completing the square, $q(x) = (x - B^{-1}a)^TB(x - B^{-1}a) + \bar{\lambda} - a^TB^{-1}a$. Therefore, (2.4) and the following problem share the same optimal solutions:

$$\begin{aligned} r_G^2 := \quad & \min \quad r^2(x) := (x - B^{-1}a)^TB(x - B^{-1}a) \\ & \text{s.t.} \quad \|x\| = 1, \end{aligned} \tag{2.8}$$

The level curves $r^2(x) = r^2$, for r a fixed non-negative constant, are the boundaries of the ellipsoids E_r, where

$$E_r := \{x = rB^{-1/2}u + B^{-1}a \mid \|u\| \leq 1\}. \tag{2.9}$$

The volume of each of these ellipsoids, where the unit is taken to be the volume of the unit ball in $\mathbb{R}^n$, is the determinant of $rB^{-1/2}$ and their center, $B^{-1}a$, is by (2.7b) in the interior of the unit ball. Therefore, problem (2.8) is equivalent to finding the largest volume ellipsoid, say E_{r_G}, among the ellipsoids E_r contained in the unit ball. Equivalence is in the sense that E_{r_G} intersects the unit sphere at an optimal x^* for (2.4). This is illustrated in Figure 2.1.

The constraint imposed on E_r to be contained in the unit ball may be modeled by a semidefinite constraint as indicated by the following lemma [2, 4]:

Lemma 2.1. *An ellipsoid*

$$E = E(Z, z) := \{x = Zu + z \mid \|u\| \leq 1\}, \qquad Z \in \mathbb{M}^{n,m}$$

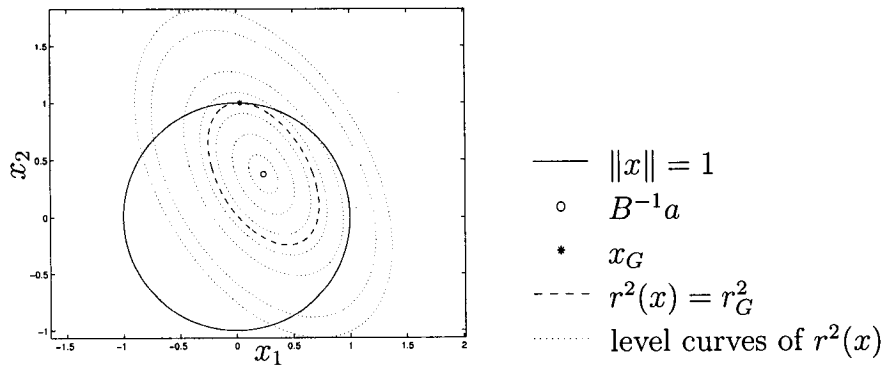

Figure 2.1: This figure illustrates problem (2.8) is equivalent to finding the largest volume ellipsoid E_{r_G} contained in the unit ball. A global optimum x_G lies in the intersection of E_{r_G} and the unit sphere.

is contained in the ellipsoid

$$W = W(Y, y) := \{x \mid (x - y)^T Y Y^T (x - y) \leq 1\}, \qquad Y \in \mathbb{M}^{n,n}\ , det(Y) \neq 0$$

if and only if there exists γ *such that*

$$\begin{pmatrix} I & Y(z-y) & YZ \\ (z-y)^T Y^T & 1-\gamma & 0 \\ Z^T Y^T & 0 & \gamma I \end{pmatrix} \succeq 0. \qquad \square \tag{2.10}$$

Applying Lemma 2.1 with $Z = rB^{-1/2}$, $z = B^{-1}a$, $Y = I$ and $y = 0$, and multiplying the matrix in (2.10) from the right and the left by a well chosen block diagonal matrix, we obtain the linear semidefinite program

$$\begin{aligned} r_G = \quad & \max_{r,\gamma} \ r \\ \text{s.t.} \quad & \begin{pmatrix} I & B^{-1}a & rI \\ a^T B^{-1} & 1-\gamma & 0 \\ rI & 0 & \gamma B \end{pmatrix} \succeq 0, \end{aligned} \tag{2.11}$$

whose optimal value is r_G. The next step is to show (2.11) may be solved by minimizing the single variable function f defined in (8). To do so, we will need the following

lemma on the Schur complement which is a consequence of Sylvester's law of inertia.

Lemma 2.2. *Let*

$$M = \begin{pmatrix} N & C^T \\ C & D \end{pmatrix}$$

be a symmetric matrix with $k \times k$ block N and $g \times g$ block D. Assume that N is positive definite. Then $M \succeq 0$ ($\succ 0$) if and only if the matrix $D - CN^{-1}C^T \succeq 0$ ($\succ 0$). □

By feasibility of (2.11), the top left square matrix of size $n+1$ is positive semidefinite and applying Lemma 2.2 with $N = I$ gives $0 \leq \gamma \leq 1 - \|B^{-1}a\|^2 < 1$ (note the left most inequality follows from our assumption $a \neq 0$). Note also r is bounded, since, for E_r to lie inside the unit ball of volume 1, $\det(rB^{-1/2}) \leq 1$ must hold. Lemma 2.2 may again be used to gain further information by applying it to the full constraint matrix in (2.11), with this time N equal to the bottom right matrix of size $n+1$ and assuming $\gamma < 1$. This gives

$$I_n - \frac{r^2}{\gamma} B^{-1} - \frac{1}{1-\gamma} B^{-1} a a^T B^{-1} \succeq 0. \tag{2.12}$$

Multiplying the left hand side of (2.12) from the left and the right by $(\gamma B)^{1/2}$ yields $r \leq \sqrt{\gamma \lambda_1(B(\gamma))}$, where $B(\gamma)$ is defined in (8) with $\Delta = 1$. Hence, for a fixed $\gamma < 1$, the largest possible r for which the matrix in (2.11) stays positive semidefinite is $r = \sqrt{\gamma \lambda_1(B(\gamma))}$. Thus to solve (2.11), one needs to find γ^* which solves

$$\begin{aligned} r_G^2 = \quad & \max \quad f(\gamma) = \gamma \lambda_1(B(\gamma)) \\ & s.t. \quad \gamma < 1. \end{aligned} \tag{2.13}$$

The rest of this chapter is concerned with solving (2.13) and constructing an optimal solution to (2.4) from γ^*. As already mentioned in Section §2.1, a solution to (2.4) is a solution for (2.1) (with $\Delta = 1$), unless a strict unconstrained minimizer of $q(x)$ lies in the interior of the unit ball. We will show in Section §2.4.2 how we may detect the existence of such a minimizer while solving (2.13).

Problem (2.13) is derived from our geometric interpretation of problem (2.8) in terms of finding a largest volume ellipsoid E_r contained in another ellipsoid, the unit ball. However, this interpretation of the problem does not appear explicitly in the algorithms we shall use to solve problem (2.13), i.e. it is only used as a modeling tool to obtain the latter problem. Maximizing the volume of ellipsoids contained in another ellipsoid appears in other contexts as well. For example, it is used in [52] to derive a measure used to preserve current Hessian information in Quasi-Newton methods.

2.3 Eigenvalue Functions

In this section, we intend to study thoroughly the two functions $\lambda_1(B(\gamma))$ and $f(\gamma)$. The former function may be evaluated, at a given value of γ, by finding the smallest value of λ among the couples (λ, v) satisfying the equation

$$\left(B^2 - \frac{1}{1-\gamma} aa^T\right) v = \lambda B v. \tag{2.14}$$

The latter equation is commonly referred to as a generalized eigenvalue problem and, for a solution (λ, v), λ is a generalized eigenvalue and v is a corresponding generalized eigenvector. It may be used to evaluate $\lambda_1(B(\gamma))$, since it can be rewritten as $B(\gamma)B^{1/2}v = \lambda B^{1/2}v$. Hence, the generalized eigenvalues of the generalized eigenvalue problem (2.14) are the same as the eigenvalues of $B(\gamma)$ and if v solves (2.14) for a generalized eigenvalue λ, then $B^{1/2}v$ is an eigenvector of $B(\gamma)$ for the same eigenvalue. We shall denote by $v(\gamma)$ a solution with unit norm to (2.14) for $\lambda = \lambda_1(B(\gamma))$. We omit the argument, i.e. we denote $v(\gamma)$ by v, whenever the context is clear. Note we do not claim $v(\gamma)$ is unique. However, when the multiplicity of $\lambda_1(B(\gamma))$ is one, there are only two possibilities which are opposite in sign.

For a given value of γ, we do not directly solve (2.14) in our algorithm to obtain the required couple $(\lambda_1(B(\gamma)), v(\gamma))$. Since the vector $B^{-1}a$ is readily available, using

a conjugate gradient method, we equivalently find the smallest eigenvalue of the (non symmetric) matrix $B - 1/(1-\gamma)B^{-1}aa^T$ and a corresponding eigenvector. This is made possible through the use of a ARPACK [24, 25, 43] subroutine, which only requires the matrix-vector multiplications $w \leftarrow (B - 1/(1-\gamma)B^{-1}aa^T)u$. This will make it possible for our algorithm to fully exploit the sparsity of the matrix B and thus, the sparsity of the matrix A.

Let $B = QDQ^T$ be an orthonormal diagonalization of B, i.e. the columns of Q are orthonormal eigenvectors of B and D is a diagonal matrix with the eigenvalues of B on its diagonal ordered increasingly such that $D_{11} = \lambda_1(B)$. Notice $\lambda_1(B) = -\bar{\lambda}$. We denote by q_j the j-th column of Q and we define i to be the multiplicity of $\lambda_1(B)$, i.e. $\lambda_1(B) = \lambda_2(B) = \ldots = \lambda_i(B) < \lambda_{i+1}(B)$. Note from the definition of B that $A_r = Q(D + \bar{\lambda}I)Q^T$, i.e. the columns of Q are eigenvectors of A_r and $\lambda_1(A_r)$, which is zero, has multiplicity i. Thus, from Table 2.1,

$$q_j^T a = 0 \qquad \text{for } j = 1, \ldots, i \tag{2.15}$$

in the hard case and we assume without loss of generality $q_1^T a \neq 0$ in the easy case.

To study the eigenvalues of $B(\gamma)$, we shall need a result which is a corollary of Weyl's inequalities [50] (see also [3, 23]).

Theorem 2.1. *Let H_1 and H_2 be two $n \times n$ symmetric matrices for which $H_1 - H_2 \succeq 0$. If the rank of $H_1 - H_2$ is r, then*

1. $\lambda_i(H_1) \geq \lambda_i(H_2)$ *for* $i = 1, 2, \ldots, n$,

2. $\lambda_i(H_1) \leq \lambda_{i+r}(H_2)$ *for* $i = 1, 2, \ldots, n - r$. □

Interlacing of the eigenvalues of $B(\gamma)$ and B follows from the theorem. Another proof of this result is given in [51] using Cauchy's inequalities [5], Lemma 3.11 in this thesis.

Corollary 2.1. *For $\gamma \neq 1$, the eigenvalues of $B(\gamma)$ and B interlace.*

1. *For $\gamma > 1$,*

$$\lambda_1(B) \leq \lambda_1(B(\gamma)) \leq \lambda_2(B) \leq \lambda_2(B(\gamma)) \leq \ldots \leq \lambda_n(B) \leq \lambda_n(B(\gamma)).$$

2. *For $\gamma < 1$,*

$$\lambda_1(B(\gamma)) \leq \lambda_1(B) \leq \lambda_2(B(\gamma)) \leq \lambda_2(B) \leq \ldots \leq \lambda_n(B(\gamma)) \leq \lambda_n(B).$$

Proof. We prove the first item, the proof of the second item being similar. Let $H_1 := B(\gamma)$ and $H_2 := B$ in Theorem 2.1 and note

$$H_1 - H_2 = -\frac{1}{1-\gamma} B^{-1/2} a a^T B^{-1/2}$$

is a rank 1 positive semidefinite matrix. □

Let $\lambda_s(B)$ be the smallest eigenvalue of B such that $a \not\perp \mathcal{N}(B - \lambda_s(B)I) = \mathcal{N}(A_r - \lambda_s(A_r)I)$. Hence, $q_j^T a = 0$, for $j = 1, \ldots, s-1$, and $q_s^T a \neq 0$. Let $\bar{D}$ be the diagonal matrix with the $n - s + 1$ largest eigenvalues of B on its diagonal ordered increasingly such that $\bar{D}_{11} = \lambda_s(B)$. Note $D = \bar{D}$ and $s = 1$ in the easy case. Define $\bar{a} := \begin{pmatrix} q_s^T a, & \ldots & , q_n^T a \end{pmatrix}^T$. Since $B^{-1/2} = Q D^{-1/2} Q^T$, we obtain

$$B(\gamma) = Q \left(\begin{array}{ccc|c} \lambda_1(B) & & 0 & \\ & \ddots & & 0 \\ 0 & & \lambda_{s-1}(B) & \\ \hline & 0 & & \bar{D} - \frac{1}{1-\gamma} \bar{D}^{-1/2} \bar{a} \bar{a}^T \bar{D}^{-1/2} \end{array} \right) Q^T. \tag{2.16}$$

Analogically to $B(\gamma)$, define $\bar{D}(\gamma) := \bar{D} - \frac{1}{1-\gamma} \bar{D}^{-1/2} \bar{a} \bar{a}^T \bar{D}^{-1/2}$. We have thus proved the following lemma, which is illustrated in Figure 2.2

Lemma 2.3. $\lambda_1(B(\gamma)) = \min(\lambda_1(B), \lambda_1(\bar{D}(\gamma)))$. □

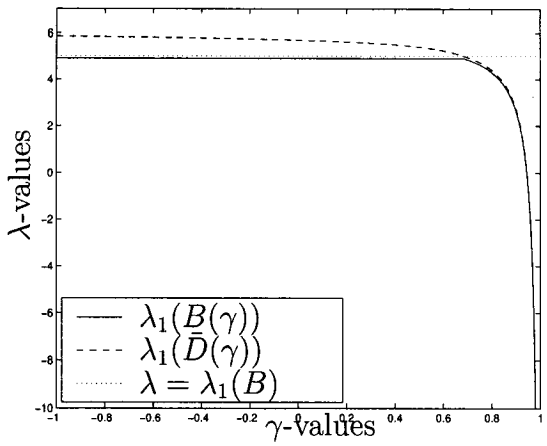

Figure 2.2: This figure illustrates Lemma 2.3

The next lemma investigates the function $\lambda_1(\bar{D}(\gamma))$.

Lemma 2.4. *$\lambda_1(\bar{D}(\gamma))$ is a concave, differentiable and strictly decreasing function for $\gamma \in (-\infty, 1)$. Furthermore, the multiplicity of $\lambda_1(\bar{D}(\gamma))$ is one for all $\gamma \in (-\infty, 1)$, $\lambda_1(\bar{D}(\gamma)) < \lambda_s(B)$, $\lim_{\gamma \to -\infty} \lambda_1(\bar{D}(\gamma)) = \lambda_s(B)$ and $\lim_{\gamma \to 1^-} \lambda_1(\bar{D}(\gamma)) = -\infty$.*

Proof. Let $\gamma \in (-\infty, 1)$ and $\lambda < \lambda_1(\bar{D})$. Then

$$\begin{aligned}
\det(\bar{D}(\gamma) - \lambda I) &= \det\left((\bar{D} - \lambda I)\left(I - \frac{1}{1-\gamma}(\bar{D} - \lambda I)^{-1}\bar{D}^{-1/2} a a^T \bar{D}^{-1/2}\right)\right), \\
&= \prod_{j=1}^{n}(\lambda_j(\bar{D}) - \lambda)\left(1 - \frac{1}{1-\gamma}\sum_{j=1}^{n-s+1}\frac{\bar{a}_j^2}{\lambda_j(\bar{D})(\lambda_j(\bar{D}) - \lambda)}\right), \\
&= \prod_{j=1}^{n}(\lambda_j(\bar{D}) - \lambda)\left(1 - \frac{1}{1-\gamma} d(\lambda)\right), \qquad (2.17)
\end{aligned}$$

where the second equality follows from Golub and Van Loan [15] and where we define,

$$d(\lambda) := \sum_{j=1}^{n-s+1} \frac{\bar{a}_j^2}{\lambda_j(\bar{D})(\lambda_j(\bar{D}) - \lambda)}.$$

From the choice of s, $\bar{a}_1 \neq 0$ and thus $\lim_{\lambda \to \lambda_1(\bar{D})^-} d(\lambda) = \infty$. Clearly $\lim_{\lambda \to -\infty} d(\lambda) = 0$ also holds. In addition, a short computation reveals $d(\lambda)$ strictly increases on

$(-\infty, \lambda_1(\bar{D}))$. Thus, by (2.17), $\lambda_1(\bar{D}(\gamma))$ uniquely solves on the latter interval

$$d(\lambda) = 1 - \gamma \tag{2.18}$$

and $\lambda_1(\bar{D}(\gamma)) < \lambda_1(\bar{D}) = \lambda_s(B)$. Monotonicity, concavity and differentiability of $\lambda_1(\bar{D}(\gamma))$ follow by implicitly differentiating (2.18). The limiting behaviors of $\lambda_1(\bar{D}(\gamma))$ also follow from this equation. Similarly to Corollary 2.1, $\lambda_1(\bar{D}) \leq \lambda_2(\bar{D}(\gamma))$ and thus $\lambda_1(\bar{D}(\gamma))$ has multiplicity one. □

From Lemmas 2.3 and 2.4, it can be observed $\lambda_1(B(\gamma))$ is the maximum of two concave curves, one of which is constant. The curves are themselves differentiable, but may intersect when $\lambda_1(B(\gamma)) = \lambda_1(B)$. Thus $\lambda_1(B(\gamma))$ may be non-differentiable at this point. However, we also observe, when $s \leq i$, that $\lambda_1(B(\gamma)) = \lambda_1(\bar{D}(\gamma))$ and that in this case it is an everywhere differentiable function. The next lemma builds on these observations and links the structure of $\lambda_1(B(\gamma))$ to the easy and hard case.

Lemma 2.5. *1. In the easy case, for $\gamma \in (-\infty, 1)$, $\lambda_1(B(\gamma))$ is a concave, everywhere differentiable and strictly decreasing function, $\lambda_1(B(\gamma)) < \lambda_1(B)$ and the multiplicity of $\lambda_1(B(\gamma))$ is one.*

2. In the hard case, for $\gamma \in (-\infty, 1)$, $\lambda_1(B(\gamma))$ is a concave function. Moreover, let $\hat{\gamma} \in (-\infty, 1)$ be the unique solution to $\lambda_1(\bar{D}(\gamma)) = \lambda_1(B)$, then $\lambda_1(B(\gamma))$ is differentiable for $\gamma \in (-\infty, 1) \setminus \hat{\gamma}$ and is non-differentiable at $\hat{\gamma}$. Define $u(\gamma) := Q \begin{pmatrix} 0, & \dots & ,0, & \bar{u}(\gamma) \end{pmatrix}^T$, where $\bar{u}(\gamma)$ is a unit eigenvector of the smallest eigenvalue of $\bar{D}(\gamma)$.

(a) For $\gamma \in (-\infty, \hat{\gamma})$, $\lambda_1(B(\gamma)) = \lambda_1(B)$ with multiplicity i and $\{q_1, \dots, q_i\}$ are i orthonormal corresponding eigenvectors.

(b) For $\gamma = \hat{\gamma}$, $\lambda_1(B(\gamma)) = \lambda_1(B)$ with multiplicity $i+1$ and $\{q_1, \dots, q_i, u(\hat{\gamma})\}$ are $i+1$ orthonormal corresponding eigenvectors.

(c) *For $\gamma \in (\hat{\gamma}, 1)$, $\lambda_1(B(\gamma)) < \lambda_1(B)$ with multiplicity one, $u(\gamma)$ is a corresponding unit eigenvector and $\lambda_1(B(\gamma))$ is a strictly decreasing function on this open interval.*

Proof. 1. The proof follows from Lemma 2.4, since the easy case implies $D = \bar{D}$ and $s = 1$.

2. The hard case implies $\lambda_1(B) < \lambda_s(B)$. Thus, by Lemma 2.4, there exists a unique $\hat{\gamma} \in (-\infty, 1)$ such that $\lambda_1(\bar{D}(\hat{\gamma})) = \lambda_1(B)$ and, by Lemma 2.3,

$$\lambda_1(B(\gamma)) = \lambda_1(B) \quad \text{for } \gamma \in (-\infty, \hat{\gamma}], \tag{2.19a}$$

$$\lambda_1(B(\gamma)) < \lambda_1(B) \quad \text{for } \gamma \in (\hat{\gamma}, 1). \tag{2.19b}$$

Concavity of the function $\lambda_1(B(\gamma))$ follows from Lemma 2.3 which shows it is the minimum of two concave functions. Differentiability is obvious for $\gamma \in (-\infty, \hat{\gamma})$, by (2.19a). For $\gamma \in (\hat{\gamma}, 1)$, since (2.19b) and Lemma 2.3 yield $\lambda_1(B(\gamma)) = \lambda_1(\bar{D}(\gamma))$, differentiability in this case follows from Lemma 2.4. As we now show, the multiplicity of the eigenvalue $\lambda_1(B(\gamma))$ changes at $\hat{\gamma}$. Therefore, loss of differentiability occurs at this value.

(a) Equations (2.16), (2.19a) and $s-1 \geq i$, implies the multiplicity of $\lambda_1(B(\gamma))$ is i. By (2.15), $a^T B^{-1/2} q_j = 0$, for $j = 1, \ldots, i$, and thus

$$B(\gamma) q_j = \lambda_1(B) q_j \qquad \text{for } j = 1, \ldots, i. \tag{2.20}$$

This shows $\{q_1, \ldots, q_i\}$ are i orthonormal eigenvectors for $\lambda_1(B(\gamma))$.

(b) Since the smallest eigenvalue of $\bar{D}(\hat{\gamma})$ is $\lambda_1(B)$ with multiplicity one and $s - 1 \geq i$, and because of (2.16) and (2.19a), we deduce the multiplicity of

$\lambda_1(B(\hat{\gamma}))$ is $i+1$. We also have

$$\begin{aligned} B(\hat{\gamma})u(\hat{\gamma}) &= Q\Big(\ 0,\ \ldots\ ,0,\ \bar{D}(\hat{\gamma})\bar{u}(\hat{\gamma})\ \Big)^T, \\ &= Q\Big(\ 0,\ \ldots\ ,0,\ \lambda_1(\bar{D}(\hat{\gamma}))\bar{u}(\hat{\gamma})\ \Big)^T, \\ &= \lambda_1(B)u(\hat{\gamma}), \end{aligned} \tag{2.21}$$

where the first equality follows from (2.16). Equations (2.20) and (2.21) imply $\{q_1, \ldots, q_i, u(\hat{\gamma})\}$ are $i+1$ orthonormal eigenvectors for the smallest eigenvalue of $B(\hat{\gamma})$.

(c) When $\gamma \in (\hat{\gamma}, 1)$, as mentioned above, $\lambda_1(B(\gamma)) = \lambda_1(\bar{D}(\gamma))$. Since $\lambda_1(\bar{D}(\gamma))$ has multiplicity one, by (2.16), the same holds for $\lambda_1(B(\gamma))$. Similarly as in the proof of item 2(b), $u(\gamma)$ is a unit eigenvector for the smallest eigenvalue of $\lambda_1(B(\gamma))$. The fact that $\lambda_1(B(\gamma))$ strictly decreases on this interval follows from Lemma 2.4.

□

We are now in position to prove the main theorem of this section concerning the function $f(\gamma)$. In particular, we show problem (2.13) is convex.

Theorem 2.2. *f is concave for $\gamma \in [0, 1)$ and has a unique optimum γ^*. Unless the hard case holds and $\gamma = \hat{\gamma}$, its derivative is given by*

$$f'(\gamma) = \lambda_1(B(\gamma)) - \frac{\gamma}{v^T B v}\left(\frac{a^T v}{1-\gamma}\right)^2, \tag{2.22}$$

where $v = v(\gamma)$ solves (2.14). In the hard case, when $\gamma = \hat{\gamma}$, let $u(\hat{\gamma})$ be defined as in Lemma 2.5 and define $\hat{v} := (1/\|B^{-1/2}u(\hat{\gamma})\|)B^{-1/2}u(\hat{\gamma})$. Then, the directional derivatives from the left and right are respectively

$$f'(\hat{\gamma}^-) = \lambda_1(B) \quad \textit{and} \quad f'(\hat{\gamma}^+) = \lambda_1(B) - \frac{\hat{\gamma}}{\hat{v}^T B \hat{v}}\left(\frac{a^T \hat{v}}{1-\hat{\gamma}}\right)^2. \tag{2.23}$$

Proof. We prove concavity directly. Let $0 \leq \mu \leq 1$, $0 \leq \gamma_i < 1$ for $i = 1, 2$, $\gamma_1 \neq \gamma_2$ and define $C(\gamma) := \gamma B(\gamma)$, then

$$\begin{aligned} f(\mu\gamma_1 + (1-\mu)\gamma_2) &= \lambda_1 \left(C(\mu\gamma_1 + (1-\mu)\gamma_2)\right) = \min_{\|y\|=1} y^T C(\mu\gamma_1 + (1-\mu)\gamma_2) y \\ &\geq \min_{\|y\|=1} y^T \left((\mu\gamma_1 + (1-\mu)\gamma_2)B - \left(\mu\tfrac{\gamma_1}{1-\gamma_1} + (1-\mu)\tfrac{\gamma_2}{1-\gamma_2}\right) B^{-1/2} a a^T B^{-1/2}\right) y \\ &\geq \mu \min_{\|y\|=1} y^T C(\gamma_1) y + (1-\mu) \min_{\|y\|=1} y^T C(\gamma_2) y = \mu\, f(\gamma_1) + (1-\mu)\, f(\gamma_2). \end{aligned} \tag{2.24}$$

The first equality follows from $\mu\gamma_1 + (1-\mu)\gamma_2 \geq 0$ and the first inequality follows since $\gamma/(1-\gamma)$ is strictly convex over $(-\infty, 1)$. Observe this inequality holds strictly if the easy case holds or if $\gamma_i \geq \hat{\gamma}$, $i = 1, 2$, if the hard case holds. The reason is that for these cases a solution y^* to $\min_{\|y\|=1} y^T C(\mu\gamma_1 + (1-\mu)\gamma_2) y$ satisfies $a^T B^{-1/2} y \neq 0$, since $\lambda_1(B(\mu\gamma_1 + (1\mu)\gamma_2) < \lambda_1(B)$. This implies f is strictly concave in the easy case for $\gamma \in [0, 1)$ or on the interval $(\max\{0, \hat{\gamma}\}, 1)$ in the hard case. Since $\lim_{\gamma \to 1^-} \lambda_1(B(\gamma)) = -\infty$, then $\lim_{\gamma \to 1^-} f(\gamma) = -\infty$. Furthermore, in the hard case, by item 2(a) of Lemma 2.5, f is an increasing linear function on $(-\infty, \hat{\gamma}]$, which implies $\gamma^* \in [\hat{\gamma}, 1)$. These observations yield uniqueness of γ^*. We now prove (2.22) when the multiplicity of $\lambda_1(B(\gamma))$ is one. Note from the discussion around equation (2.14) that $1/\|B^{1/2}v\| B^{1/2}v$ is a unit eigenvector for the smallest eigenvalue of $B(\gamma)$. Therefore (see e.g. Horn and Johnson [23]),

$$\begin{aligned} f'(\gamma) &= \lambda_1(B(\gamma)) - \gamma \left(\frac{B^{1/2}v}{\|B^{1/2}v\|}\right)^T \left(\frac{1}{(1-\gamma)^2} B^{-1/2} a a^T B^{-1/2}\right) \left(\frac{B^{1/2}v}{\|B^{1/2}v\|}\right), \\ &= \lambda_1(B(\gamma)) - \frac{\gamma}{v^T B v} \left(\frac{a^T v}{1-\gamma}\right)^2. \end{aligned}$$

When the multiplicity of $\lambda_1(B(\gamma))$ is not one, but f is differentiable, we conclude from Lemma 2.5 $\gamma \in (-\infty, \hat{\gamma})$ and the hard case holds. In this case $\lambda_1(B(\gamma)) = \lambda_1(B)$ has multiplicity i and, trivially, $f'(\gamma) = \lambda_1(B)$. Note for any v which satisfies (2.14), we have, as mentioned above, that $B^{1/2}v$ is an eigenvector for the smallest eigenvalue of

$B(\gamma)$. By item 2(a) of Lemma 2.5, $B^{1/2}v \in \text{span}\{q_1, \ldots, q_i\}$ and therefore

$$v \in \text{span}\{q_1, \ldots, q_i\}. \tag{2.25}$$

Thus, by (2.15), $v^T a = 0$, proving (2.22) also holds in this case. Finally we are left to prove (2.23). The left derivative in $\hat{\gamma}$ clearly follows from (2.19a). Recall, from Lemma 2.4, $\lambda_1(\bar{D}(\hat{\gamma}))$ has multiplicity one. Thus, as proved above, and using Lemma 2.3, we have

$$f'(\hat{\gamma}^+) = \frac{d\,\gamma\lambda_1(\bar{D}(\gamma))}{d\,\gamma}\bigg|_{\gamma=\hat{\gamma}} = \lambda_1(\bar{D}(\hat{\gamma})) - \frac{\hat{\gamma}}{\bar{u}(\hat{\gamma})^T\bar{u}(\hat{\gamma})}\left(\frac{\bar{a}^T\bar{D}^{-1/2}\bar{u}(\hat{\gamma})}{1-\hat{\gamma}}\right)^2. \tag{2.26}$$

All that is left to note is that (2.26) is exactly the formula for $f'(\hat{\gamma}^+)$ in (2.23). □

Figures 2.3, 2.4 and 2.5 illustrate the function f for different cases.

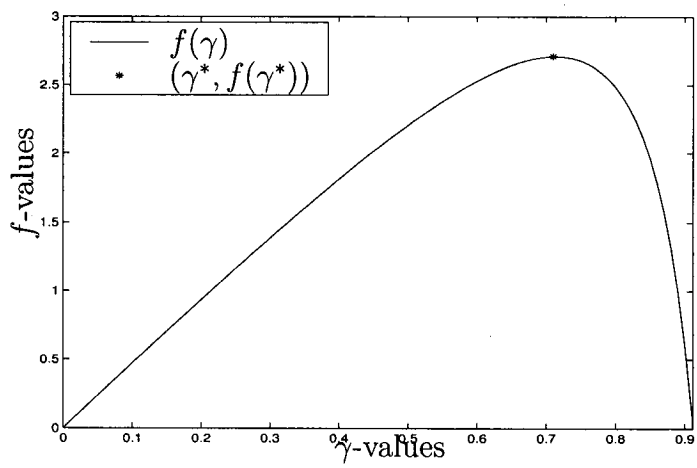

Figure 2.3: $f(\gamma)$ in the easy case

The previous theorem implies we can obtain an approximate value of r_G^2 to any desired precision, since all is needed is to maximize the concave function f on the closed interval $[0, 1 - \|B^{-1}a\|^2]$ (better bounds are given in Section §2.6.2). Note the two end points are the roots of f, since $B^{-3/2}a \in \mathcal{N}(B(1 - \|B^{-1}a\|^2))$ and $\lambda_1(B(\gamma)) \geq 0$ for $\gamma < 1 - \|B^{-1}a\|^2$ (apply Lemma 2.2 to the top left square matrix in (2.11) of size $n+1$).

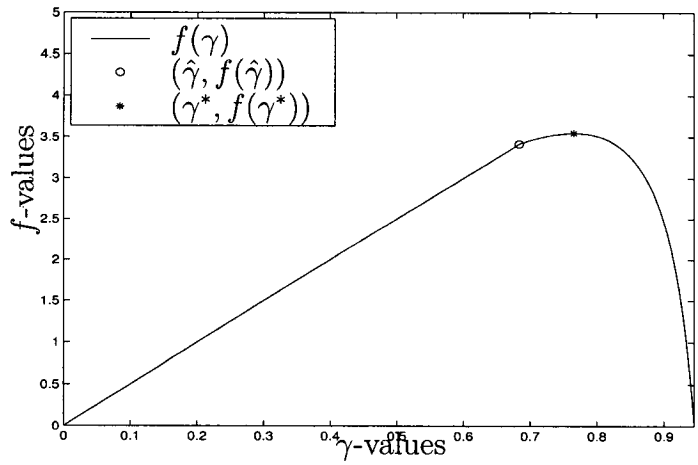

Figure 2.4: $f(\gamma)$ in the hard case (case 1)

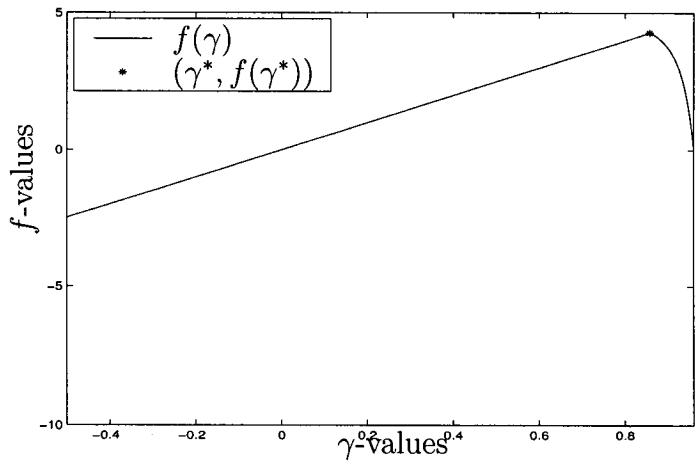

Figure 2.5: $f(\gamma)$ in the hard case (case 2)

In Sections §2.5 and §2.6 we specify how we solve (2.13). However, even if the optimal γ^* is obtained, it is not clear yet how the optimal x^* for (2.4) and its corresponding Lagrange multiplier λ^* may be recovered.

2.4 Constructing an Optimal Solution

This section is comprised of two parts: In Section §2.4.1, we focus on solving (2.4) from the information gained by solving the convex problem (2.13). As already mentioned in Section 2.1, a solution to (2.4) is a solution to the TRS (2.1) unless $A \succ 0$ and $\|A^{-1}a\| < 1$. In Section §2.4.2, we show that, while solving (2.13), we are able to verify if these two conditions hold and compute, in the affirmative case, the unconstrained minimizer $A^{-1}a$.

2.4.1 Solving TRS$_=$

We show how to obtain an optimal solution for problem (2.2). However, since x^* is also optimal for problem (2.4), we still focus on the latter problem. To construct a couple (x^*, λ^*) which satisfies (2.5) from an optimal γ^*, we need a relation between the variables γ and x. A clever observation reveals we have almost explicitly written it down already. Namely rewriting (2.14), we obtain $v(\gamma)$ solves

$$(B - \lambda_1(B(\gamma))I)\left(\frac{1-\gamma}{a^T v(\gamma)}\right) Bv(\gamma) = a, \tag{2.27}$$

when $a^T v(\gamma) \neq 0$. Now observe $B - \lambda_1(B(\gamma))I \succeq 0$ and notice this matrix may be written as $A_r - (\lambda_1(B(\gamma)) + \bar{\lambda})I$. The optimality conditions (2.5) suggests defining

$$x(\gamma) := \left(\frac{1-\gamma}{a^T v(\gamma)}\right) Bv(\gamma). \tag{2.28}$$

We should be careful with this definition, since it is valid only if $a^T v(\gamma) \neq 0$ and since $v(\gamma)$ is not uniquely defined. However, $x(\gamma)$ should be uniquely defined. As we will see, unless the hard case holds and $\gamma \leq \hat{\gamma}$, $x(\gamma)$ is well defined. If in addition $\|x(\gamma)\| = 1$, then (2.5) implies it is also optimal for (2.4). More generally, the following lemma shows $x(\gamma)$ is the optimum to a well-chosen problem.

Lemma 2.6. *Let $\gamma < 1 - \|B^{-1}a\|^2$ and assume if the hard case holds that $\gamma > \hat{\gamma}$. Let $x(\gamma)$ be defined as in (2.28). Then $a^T v(\gamma) \neq 0$ and $x(\gamma)$ solves*

$$\begin{aligned} \min \quad & (x - B^{-1}a)^T B(x - B^{-1}a) \\ \text{s.t.} \quad & \|x\| \geq \|x(\gamma)\| \end{aligned} \tag{2.29}$$

with corresponding Lagrange multiplier $\lambda_1(B(\gamma))$.

Proof. For simplicity, let $v = v(\gamma)$. Note first $a^T v \neq 0$, otherwise, by (2.14), $\lambda_1(B(\gamma))$ is an eigenvalue of B and this contradicts items 1 and 2(c) of Lemma 2.5, which yield $\lambda_1(B(\gamma)) < \lambda_1(B)$, unless the hard case holds and $\gamma \leq \hat{\gamma}$. Therefore, $x(\gamma)$ is well defined. Similarly to (2.5), the necessary and sufficient optimality conditions for (2.29) imply we need to prove

$$(B - \lambda_1(B(\gamma))I)x(\gamma) = a, \tag{2.30a}$$

$$B - \lambda_1(B(\gamma))I \succeq 0, \tag{2.30b}$$

$$\lambda_1(B(\gamma)) \geq 0. \tag{2.30c}$$

Conditions (2.30a) follows from (2.27) and (2.30b) holds again by $\lambda_1(B(\gamma)) < \lambda_1(B)$. From the discussion at the end of Section §2.3, $1 - \|B^{-1}a\|^2$ is the root of the decreasing function $\lambda_1(B(\gamma))$, and thus $\lambda_1(B(\gamma)) \geq 0$ for $\gamma \leq 1 - \|B^{-1}a\|^2$, proving (2.30c). □

To find an optimal γ^* for problem (2.4), Theorem 2.2 suggests solving $f'(\gamma) = 0$. On the other hand, Lemma 2.6 suggests solving $\|x(\gamma)\| = 1$. Hence, it is not a surprise these two functions are linked.

Lemma 2.7. *Assume the conditions of Lemma 2.6 hold. Then*

$$\|x(\gamma)\| > (=, <)\ 1 \iff f'(\gamma) > (=, <)\ 0. \tag{2.31}$$

Proof. We have

$$\begin{aligned}
\|x(\gamma)\|^2 &= \left(\frac{1-\gamma}{a^T v}\right)^2 v^T B^2 v, \\
&= 1-\gamma+\lambda_1(B(\gamma))\left(\frac{1-\gamma}{a^T v}\right)^2 v^T B v, && \text{(2.32a)} \\
&= 1+f'(\gamma)\left(\frac{1-\gamma}{a^T v}\right)^2 v^T B v, && \text{(2.32b)} \\
&= 1+f'(\gamma)\|x(\gamma)\|^2 \frac{v^T B v}{v^T B^2 v}, && \text{(2.32c)}
\end{aligned}$$

where (2.32a) follows from (2.14) and (2.32b), from (2.22). The conclusion follows by writing (2.32c) as

$$\|x(\gamma)\|^2 = \frac{1}{1-f'(\gamma)\frac{v^T B v}{v^T B^2 v}}. \tag{2.33}$$

□

It appears almost clear from the last two lemmas how the couple (x^*, λ^*) may be recovered from an optimal γ^*. Still, a major concern is the hard case. Precisely, $x(\gamma^*)$ is not defined when f is non-differentiable at the optimum. Equation (2.6) suggests how (x^*, λ^*) should be obtained in this case.

Theorem 2.3. *Let γ^* solve* (2.13).

1. *In the easy case or in the hard case, when $\gamma^* \neq \hat{\gamma}$, let $v(\gamma^*)$ solve* (2.14), *then $a^T v(\gamma^*) \neq 0$ and*

$$x^* \equiv \frac{1-\gamma^*}{a^T v(\gamma^*)} B v(\gamma^*) \tag{2.34}$$

solves (2.4) *with Lagrange multiplier $\lambda_1(B(\gamma^*)) + \bar{\lambda}$.*

2. *In the hard case, when $\gamma^* = \hat{\gamma}$, let $\hat{v}$ be defined as in Theorem 2.2, then $a^T \hat{v} \neq 0$ and*

$$x^* \equiv \frac{1-\gamma^*}{a^T \hat{v}} B \hat{v} + \alpha z \tag{2.35}$$

solves (2.4) *with zero Lagrange multiplier, where $z \in \mathcal{N}(A_r)$ and α is chosen to satisfy $\|x^*\| = 1$.*

Proof. 1. By items 1 and 2(c) of Lemma 2.5, Theorem 2.2 and since γ^* solves (2.13), f is differentiable at γ^* and $\lambda_1(B(\gamma^*))$ has multiplicity one. Thus $f'(\gamma^*) = 0$ and, by Lemma 2.6 and Lemma 2.7, x^* solves (2.4) and $\lambda_1(B(\gamma^*))+\bar{\lambda}$ is its (unique) corresponding Lagrange multiplier. Note $B - \lambda_1(B(\gamma))I = A_r - (\lambda_1(B(\gamma^*)) + \bar{\lambda})I$ was used.

2. Assume $a^T\hat{v} = 0$. By definition of $\hat{v}$, this is equivalent to $\bar{a}^T\bar{D}^{-1/2}\bar{u}(\hat{\gamma}) = 0$. Together with the fact that $\bar{u}(\hat{\gamma})$ is an eigenvector for the smallest eigenvalue of $\bar{D}(\hat{\gamma})$, we obtain $\lambda_1(\bar{D}(\hat{\gamma}))$ is an eigenvalue of $\bar{D}$. However, this contradicts Lemma 2.4 which says $\lambda_1(\bar{D}(\gamma)) < \lambda_1(\bar{D})$, for all $\gamma < 1$. Thus, $a^T\hat{v} \neq 0$. Now, by optimality of $\hat{\gamma}$, $f'(\hat{\gamma}^+) \leq 0$, and, similarly to the proof of Lemma 2.7, we may prove, using (2.23), that $(1-\hat{\gamma})/|a^T\hat{v}|\|B\hat{v}\| \leq 1$. Thus, it is always possible to choose z and α to satisfy $\|x^*\| = 1$. This shows x^* is well defined.

   By construction, $\hat{v}$ solves

$$\left(B^2 - \frac{1}{1-\hat{\gamma}}aa^T\right)\hat{v} = \lambda_1(B(\hat{\gamma}))B\hat{v},$$

   which may be rewritten as

$$(B - \lambda_1(B(\hat{\gamma}))I)\left(\frac{1-\hat{\gamma}}{a^T\hat{v}}\right)B\hat{v} = a.$$

   By item 2(a) of Lemma 2.5, $\lambda_1(B(\hat{\gamma})) = \lambda_1(B)$ and noting $z \in \mathcal{N}(B - \lambda_1(B)I)$, we have shown

$$(B - \lambda_1(B)I)x^* = a, \tag{2.36a}$$

$$B - \lambda_1(B)I \succeq 0, \tag{2.36b}$$

$$\|x^*\| = 1. \tag{2.36c}$$

   Finally, from $B - \lambda_1(B)I = A_r$, we see the equations (2.36) are exactly the optimality conditions (2.5) for problem (2.4) with $\lambda^* = 0$.

□

Corollary 2.2. *The following statements hold for problem* (2.4).

1. *Assume the hard case holds for* (2.2)*, then the hard case 2 occurs if and only if* $\gamma^* = \hat{\gamma}$*. The hard case 2(i) occurs if and only if* $f'(\hat{\gamma}^+) = 0$.

2. *The easy case or the hard case 1 occurs if and only if* $f'(\gamma^*) = 0$.

Proof. 1. If the hard case 2 occurs, then the Lagrange multiplier λ^* for x^* is zero. Furthermore, $\gamma^* \in [\hat{\gamma}, 1)$, since f is increasing for $\gamma \in (-\infty, \hat{\gamma})$. Now, by item 2(c) of Lemma 2.5, $\gamma^* \in (\hat{\gamma}, 1)$ cannot occur, otherwise $\lambda^* = \lambda(B(\gamma^*)) + \bar{\lambda} < \lambda_1(B) + \bar{\lambda} = 0$. Hence $\gamma^* = \hat{\gamma}$. The converse statement holds from item 2 of Theorem 2.3, since $\lambda^* = 0$ when $\gamma^* = \hat{\gamma}$. Finally, when the hard case 2(i) occurs, from Table 2.1, $\|A_r^\dagger a\| = 1$. We can show $x(\hat{\gamma}) := \frac{1-\hat{\gamma}}{a^T\hat{v}} B\hat{v} = A_r^\dagger a$, using $q_j^T x(\hat{\gamma}) = 0$, $j = 1 \ldots i$. Thus $\|x(\hat{\gamma})\| = 1$. Following the lines in the proof of Lemma 2.7, we deduce this is equivalent to $f'(\hat{\gamma}^+) = 0$. Reverse these steps to obtain the converse statement.

2. The proof follows from item 1 and Theorem 2.2.

□

The latter theorem is useful in two ways. First, it suggest a way to solve (2.4) in the easy case or the hard case 1: find the root γ^* to $f'(\gamma) = 0$ and construct x^* from (2.34). Second, it shows with Lemma 2.6 that x^* solves

$$\begin{aligned} r_G^2 = \quad & \min \quad (x - B^{-1}a)^T B (x - B^{-1}a) \\ & \text{s.t.} \quad \|x\| \geq 1. \end{aligned} \tag{2.37}$$

This problem is therefore a dual of problem (2.13) and strong duality holds. In other words, if γ and x are feasible respectively for (2.13) and (2.37), then $f(\gamma) \leq r^2(x)$ and the optimal value of these two problems are equal. This allows us in our algorithm to compute a duality gap: from Lemma 2.7, if $f'(\gamma) \leq 0$, then $x = x(\gamma)$ is feasible for

(2.37) and the duality gap, i.e. the interval of uncertainty for r_G^2, is

$$r^2(x) - f(\gamma). \tag{2.38}$$

However, it is not clear how we can take advantage of (2.35) in the hard case 2. Specifically, how do we obtain the desired vector $\hat{v}$? We will see this will be made possible through our shift procedure. Practical methods to deal with the hard case is the subject of Section §2.5.

2.4.2 Solving TRS

If $A \succ 0$ and $\|A^{-1}a\| < 1$, then a solution to (2.2) or (2.4) is no longer optimal for (2.1). Our strategy is nevertheless to assume a priori the optimum of (2.1) lies on the boundary of the unit ball. As we show in this section, we are able to check implicitly if the two latter conditions hold from the information available at the iterates of our algorithm.

Recall in problem (2.4) we have reset the matrix A to A_r. This is done in our algorithm only if $\lambda_1(A) \leq 0$ for a reason to be made clear in Section §2.5.2. If this case holds, we set $\mu = \lambda_1(A)$ (to keep the smallest eigenvalue in memory) and reset $A \leftarrow A_r$. Otherwise, if $\lambda_1(A) > 0$, we do not reset A and set $\mu = 0$. Note if the latter case holds that the different cases for the equality trust-region subproblem (2.2) are obtained by replacing A by $A - \lambda_1(A)$ in Table 2.1. Making this change gives Table 2.2.

No matter if A is reset or not in our algorithm, problem (2.2) or (2.4) may be equivalently solved by solving problem (2.13). However, the choice of the parameter $\bar{\lambda}$ which defines B will be different in either cases and we shall have $B = A - (\mu + \bar{\lambda})I$.

We now show how we implicitly detect, as we are solving problem (2.13), when an unconstrained minimizer of problem (2.1) lies in the interior of the unit ball. Let $\gamma < 1$ and assume if the hard case holds that $\gamma > \hat{\gamma}$. From (2.30a) and (2.30b), if

Easy case	Hard case 1	Hard case 2
$a \not\perp \mathcal{N}(A - \lambda_1(A)I)$	$a \perp \mathcal{N}(A - \lambda_1(A)I)$	$a \perp \mathcal{N}(A - \lambda_1(A)I)$
	and	and
(implies $\lambda^* < \lambda_1(A)$)	$\lambda^* < \lambda_1(A)$	$\lambda^* = \lambda_1(A)$
		(i) $\|(A - \lambda_1(A)I)^\dagger a\| = 1$
		(ii) $\|(A - \lambda_1(A)I)^\dagger a\| < 1$

Table 2.2: The three different cases for the equality trust region subproblem (2.2)

$\lambda_1(B(\gamma)) + \bar{\lambda} + \mu > 0$, then

$$(A - (\mu + \bar{\lambda} + \lambda_1(B(\gamma)))I)x(\gamma) = a,$$
$$A - (\mu + \bar{\lambda} + \lambda_1(B(\gamma)))I \succeq 0,$$

and if $\lambda_1(B(\gamma)) + \bar{\lambda} + \mu > 0$, then $x(\gamma)$ solves

$$\begin{aligned} \min \quad & q(x) = x^T Ax - 2a^T x \\ \text{s.t.} \quad & \|x\| \geq \|x(\gamma)\| \end{aligned}$$

with Lagrange multiplier $\lambda_1(B(\gamma)) + \bar{\lambda} + \mu$. Now assume $f'(\gamma) < 0$, so that, by Lemma 2.7, $\|x(\gamma)\| < 1$. Therefore, $x(\gamma)$ minimizes q over the exterior of a ball with radius smaller than one. The sign of the Lagrange multiplier for $x(\gamma)$ implies it is also unique. Thus an unconstrained minimizer of q exists and lies in the interior of the unit ball. In particular, this implies A is positive definite. Hence, in our algorithm, if γ satisfies all the conditions mentioned, we return $A^{-1}a$, computed using a conjugate gradient method.

2.5 Handling the Hard Case

This section discusses how our algorithm deals with the hard case. As stated in Corollary 2.2, in the hard case 1, we may always solve $f'(\gamma) = 0$ to obtain γ^* and, by (2.34), x^*. In the hard case 2(i), is it also possible to obtain a nearly optimal solution, since $f'(\hat{\gamma}^+) = 0$. However, in the hard case 2(ii), we encounter some difficulty, since $f'(\gamma)$ is bounded away from zero. This is caused by $f'(\hat{\gamma}^+) < 0$. Similarly, $\|x(\gamma)\|$, for $\gamma > \hat{\gamma}$, is bounded away from 1. This is in relation with our remark at the end of Section §2.1 on $\|x(\lambda)\|$. Thus, the hard case needs to be handled without having to rely on solving either $f'(\gamma) = 0$ or $\|x(\gamma)\| = 1$.

2.5.1 Stepping to the boundary

One of the way we treat the hard case is based in flavor on [32, Lemma 3.4], which is restated later on as Lemma 2.10. Given any vector x which does not lie on the boundary of the unit ball, our intention is to move to the boundary and obtain a feasible solution for (2.4) (or (2.2)) and (2.37), where the duality gap (2.38) is decreased. The following lemma [9, 10] provides a way to achieve this goal given two points $x(\gamma)$ on each side of the boundary of the unit ball.

Lemma 2.8. *Let* $0 < \Delta_1 < 1 < \Delta_2$ *and let*

$$x_h \in \operatorname{argmin}\{r^2(x) : \|x\| \geq \Delta_1\} \tag{2.39a}$$

$$x_e \in \operatorname{argmin}\{r^2(x) : \|x\| \geq \Delta_2\} \quad . \tag{2.39b}$$

Assume $\|x_h\| = \Delta_1$, $\|x_e\| = \Delta_2$, $x_h^T(x_e - x_h) \neq 0$ *and the Lagrange multiplier* λ *for problem* (2.39a) *satisfies* $B - \lambda I \succ 0$. *Let* $m(\alpha) := r^2(x_h + \alpha(x_e - x_h))$. *Then* $m'(\alpha) \geq 0$, *for* $\alpha \in [0, 1]$, *and therefore* $r^2(x_h + \alpha(x_e - x_h)) \leq r^2(x_e)$, *for* $\alpha \in [0, 1]$. *In particular, for* $\bar{\alpha} \in [0, 1]$ *such that* $\|x_h + \bar{\alpha}(x_e - x_h)\| = 1$ *we have that* $x_h + \bar{\alpha}(x_e - x_h)$ *is feasible for* (2.37) *and has a smaller objective value than* x_e. □

Define the *hard side* and the *easy side* respectively as the side of function f where $f'(\gamma) < 0$ and $f'(\gamma) > 0$. Assume we are given two values of the variable γ, γ_h and γ_e, respectively on the hard side and the easy side, such that both are strictly greater than $\hat{\gamma}$ if the hard case holds. Then, according to Lemmas 2.6 and 2.7, $x_h := x(\gamma_h)$ and $x_e := x(\gamma_e)$ satisfy respectively (2.39a) and (2.39b), with $\Delta_1 = \|x_h\|$ and $\Delta_2 = \|x_e\|$. Note $\Delta_1 < 1 < \Delta_2$. Furthermore, by Lemma 2.6, $\lambda_1(B(\gamma_h))$ is the Lagrange multiplier for x_h and, by items 1 and 2(c) of Lemma 2.5, $B - \lambda_1(B(\gamma_h))I \succ 0$. Thus, if we assume $x_h^T(x_e - x_h) \neq 0$, the use of Lemma 2.8 is clear: let

$$\alpha = \frac{1 - \|x_h\|^2}{x_h^T(x_e - x_h) + \sqrt{(x_h^T(x_e - x_h))^2 + \|x_e - x_h\|^2(1 - \|x_h\|^2)}} \tag{2.40}$$

and define

$$x_{\text{new}} := x_h + \alpha(x_e - x_h). \tag{2.41}$$

Thus, x_{new} is feasible for (2.4) (or (2.2)) and (2.37), and $r^2(x_{\text{new}}) \leq r^2(x_e)$.

One problem with (2.41) is its inapplicability in the hard case 2. Indeed, any γ_e on the easy side satisfies $\gamma_e \leq \hat{\gamma}$. In this case we take a step to the boundary from γ_h in the direction of an eigenvector of $\lambda_1(A)$, say q_1, as suggested by (2.6) and Lemma 2.10. Namely,

$$x_{\text{new}} := x_h + \alpha q_1, \tag{2.42}$$

where

$$\alpha = \frac{1 - \|x_h\|^2}{x_h^T q_1 + \frac{x_h^T q_1}{|x_h^T q_1|}\sqrt{(x_h^T q_1)^2 + (1 - \|x_h\|^2)}}. \tag{2.43}$$

Note the choice of α in (2.43) is driven by the desire to make $r^2(x_h + \alpha q_1)$ as small as possible. There is a nice semidefinite programming duality theory based on (2.2) which reinforces the choice of q_1 as a step direction [39, pp. 279]. Using [29, Lemma 3.4], we may also show, when $\lambda_1(A) < 0$, that q_1 points in the direction of a solution of (2.1) for an infinitely large Δ. However, a simple argument for justifying a step toward q_1, is that $q(x)$ decreases along q_1 far from the origin.

The two steps to the boundary we described are designed to handle the hard case 2(ii), but note we have not assume this case holds. In fact, stepping to the boundary is quite often beneficial even in the other cases and we use (2.41) and (2.42) whenever this step decreases the duality gap (2.38).

2.5.2 Shifting the eigenvalues of A

The authors in [10] have shown problem (2.4) may be solved by solving instead an equality constrained TRS where the easy case holds. The optimal solution to this latter equality constrained TRS is either identical to the optimal solution of problem (2.4) or is equal to $(A_r)^\dagger a$ and lies within the unit ball. In both cases, we may easily recover the optimal solution to problem (2.4). The key result is that we may avoid the hard case 2(ii). In this section, we show that shifting the smallest eigenvalues of A_r, which have an eigenspace orthogonal to a, produces the desired easy case TRS mentioned above. We assume for the rest of this section the hard case holds for problem (2.4).

When $\gamma < \hat{\gamma}$, if $v \equiv v(\gamma)$ satisfies (2.14), recall (2.25) is satisfied. We may assume $v = q_1$. In our algorithm, we use v to shift eigenvalues in the spectral decomposition of B. The following lemma shows how the function $\lambda_1(B(\gamma))$ changes.

Lemma 2.9. *Let* $\tilde{B} := B + \sum_{j=1}^{s-1} \beta_j q_j q_j^T$, *where* $\beta_j \geq 0$. *Define*

$$\begin{aligned} m_1 &:= \min(\lambda_j(B) + \beta_j : j = 1, \ldots, s-1), \\ m_2 &:= \lambda_1(\bar{D}(\gamma)), \end{aligned}$$

then $\lambda_1(\tilde{B}(\gamma)) = \min(m_1, m_2)$.

Proof. Replace in (2.16) $B(\gamma)$ by $\tilde{B}(\gamma)$ and $\lambda_j(B)$ by $\lambda_j(B) + \beta_j$, $j = 1, \ldots, s-1$. The conclusion follows similarly to how Lemma 2.3 was deduced. □

Corollary 2.3. *If* $\beta_k = 0$, *for some* $k = 1, \ldots, i$, *then* $\lambda_1(\tilde{B}(\gamma)) = \lambda_1(B(\gamma))$.

Proof. In Lemma 2.9 we obtain $m_1 = \lambda_1(B)$. The conclusion follows from Lemmas 2.3 and 2.9. □

Now let $v_1 := v$, where v is the vector mentioned above. If $i = 1$, then we have constructed a basis for span$\{q_1\}$. If $i > 1$, and we reset B to $B + \beta v_1 v_1^T$, where we choose $\beta > 0$, Corollary 2.3 implies the function $\lambda_1(B(\gamma))$ has not changed. Hence, $f(\gamma)$ has not changed. What has changed is the multiplicity of $\lambda_1(B)$, which is now $i-1$. Therefore, if as previously, for $\gamma \in (-\infty, \hat{\gamma})$, we compute a unit vector $v_2 := v(\gamma)$, which satisfies (2.14), then v_2 is a unit eigenvector for $\lambda_1(B)$. We may assume $v_2 = q_2$. It is also perpendicular and linearly independent from v_1. Repeating this process i times we obtain i orthonormal eigenvectors $v_1, \ldots, v_i$, a basis for span$\{q_1, \ldots, q_i\}$, and we may assume $v_j = q_j$, $j = 1, \ldots, i$. However, doing the last perturbation

$$B \leftarrow B + \beta v_i v_i^T, \tag{2.44}$$

will change f. Namely, the function may either be everywhere differentiable or non-differentiable in $\hat{\gamma}_{\text{new}}$, a value we define shortly. If $m_1 \geq \lambda_s(B)$, f changes to $\tilde{f}(\gamma) := \gamma\lambda_1(\bar{D}(\gamma))$, for $\gamma < 1$, and is thus everywhere differentiable. Otherwise, we define $\hat{\gamma}_{\text{new}}$ as the solution to $\lambda_1(\bar{D}(\gamma)) = m_1$ and f changes to $\tilde{f}$ defined as

$$\tilde{f}(\gamma) := \begin{cases} \gamma m_1, & \text{if } \gamma \leq \hat{\gamma}_{\text{new}}, \\ \gamma\lambda_1(\bar{D}(\gamma)), & \text{if } \hat{\gamma}_{\text{new}} < \gamma < 1. \end{cases} \tag{2.45}$$

This is illustrates in Figure 2.6.

Because we choose $\beta_j > 0$ for $j = 1, \ldots, i$, then $m_1 > \lambda_1(B)$ and Lemma 2.4 implies $\hat{\gamma}_{\text{new}} < \hat{\gamma}$. Therefore, for $\gamma < \hat{\gamma}$, $\tilde{f}$ has a slope which is greater than the slope of f. Note maximizing $\tilde{f}$ over $(-\infty, 1)$ has the same optimal value as $\min\{x^T(\tilde{A})x - 2a^Tx + a^T(\tilde{B})^{-1}a - \bar{\lambda} : \|x\| = 1\}$, where $\tilde{A} := A_r + \sum_{j=1}^{i} \beta_j q_j q_j^T$ and $\tilde{B}$ is defined as in Lemma 2.9. The latter problem may not have the same optimal value as (2.2), since $\tilde{f}$ does not necessarily have the same optimum as f. From (2.45) and Corollary 2.2,

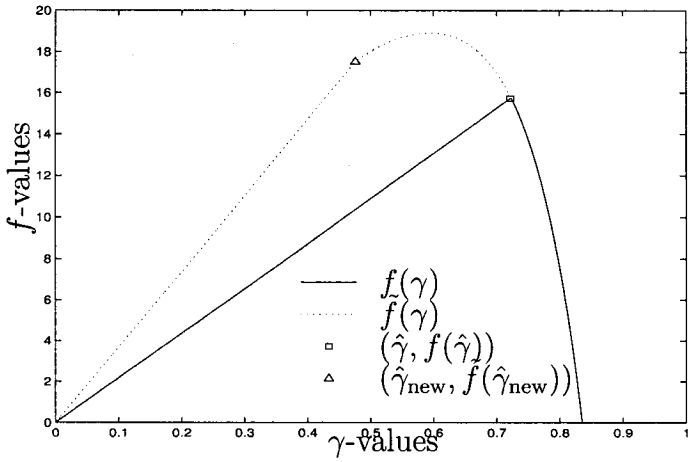

Figure 2.6: The figure illustrates how f changes to $\tilde{f}$. In the cases illustrated, $m_1 < \lambda_s(B)$.

the optimums will differ if and only if the hard case 2(ii) occurs. If this case holds and $\lambda_1(A) \leq 0$ (so that A has been reset to A_r), Table 2.1 yields $\|A_r^\dagger a\| < 1$. Hence, $A_r^\dagger a$ is the optimal solution to

$$\begin{array}{ll} \min & \tilde{q}(x) := x^T(\tilde{A})x - 2a^T x + a^T(\tilde{B})^{-1}a - \bar{\lambda} \\ \text{s.t.} & \|x\| \leq 1, \end{array} \tag{2.46}$$

since

$$\begin{aligned} A_r^\dagger a &= Q\left(\begin{array}{ccccccc} 0 & \ldots & 0 & \frac{q_{i+1}^T a}{\lambda_{i+1}(A_r)} & \cdots & \frac{q_n^T a}{\lambda_n(A_r)} \end{array}\right)^T \\ &= Q\left(\begin{array}{ccccccc} \frac{0}{\beta_1} & \cdots & \frac{0}{\beta_i} & \frac{q_{i+1}^T a}{\lambda_{i+1}(A_r)} & \cdots & \frac{q_n^T a}{\lambda_n(A_r)} \end{array}\right)^T \\ &= (A_r + \sum_{j=1}^{i} \beta_j q_j q_j^T)^{-1} a = \tilde{A}^{-1} a. \end{aligned} \tag{2.47}$$

Therefore, the optimal solution to (2.46) is the unconstrained minimizer of $\tilde{q}$. By (2.6), an optimal solution to (2.4) is recovered from

$$x^* = \tilde{A}^{-1}a + \alpha q_k, \ \lambda^* = \lambda_1(A_r) = 0, \tag{2.48}$$

where $k \in \{1, \ldots, i\}$ and where α is chosen such that $\|x^*\| = 1$. Thus the optimal solution to (2.1) and (2.2) is x^* with Lagrange multiplier $\lambda_1(A)$. Note for $\gamma \in (\hat{\gamma}_{\text{new}}, \hat{\gamma})$

such that $\tilde{f}'(\gamma) \leq 0$, $\lambda_1(\tilde{B}(\gamma)) + \bar{\lambda} > 0$ and $\|\tilde{x}(\gamma)\| < 1$, where $\tilde{x}(\gamma) = (1-\gamma)/(a^T v)\tilde{B}v$ and where v solves (2.14) with B replaced by $\tilde{B}$. Thus, the theory of Section §2.4.2 applies and we will be able to detect the optimum of (2.46) is the unconstrained minimizer of $\tilde{q}$.

As mentioned in Section §2.4.2, if $\lambda_1(A) > 0$, then we do not reset A to A_r. Recall from that section the choice of $\bar{\lambda}$ is different, but the theory of Section §2.2 is still valid. In particular, problem (2.2) may be equivalently solved through problem (2.13). Again the optimum of f and $\tilde{f}$ will differ if and only if the hard case 2(ii) (see Table 2.2) holds for problem (2.2). However, and optimal x^* for the latter problem, with Lagrange multiplier $\lambda_1(A)$, is not in this case the optimal value of (2.1), since the Lagrange multiplier λ^* at an optimal solution satisfies $\lambda^* \leq 0$. Hence, the solution to problem (2.1) in this case is the unconstrained minimizer of q which lies in the interior of the unit ball, i.e. $\|A^{-1}a\| < 1$ and its Lagrange multiplier is zero. Now $A^{-1}a$ is the optimal solution of problem (2.46), with $\tilde{A} := A + \sum_{j=1}^{i} \beta_j q_j q_j^T$. Similarly to (2.47) we have $A^{-1}a = \tilde{A}^{-1}a$. Again, for $\gamma \in (\hat{\gamma}_{\text{new}}, \bar{\gamma})$, where $\bar{\gamma}$ solves $\lambda_1(B(\gamma)) + \bar{\lambda} = 0$, and for γ such that $\tilde{f}'(\gamma) \leq 0$, we have $\lambda_1(\tilde{B}(\gamma)) + \bar{\lambda} > 0$ and $\|\tilde{x}(\gamma)\| < 1$. We will thus be able to detect the optimum of (2.46) is the unconstrained minimizer of $\tilde{q}$, which is also in this case the unconstrained minimizer of q.

We may always make (2.46) an easy case TRS by choosing the β_j, $j = 1, \ldots, s-1$, large enough so that $m_1 \geq \lambda_s(B)$ holds. As we have shown, in this case, (2.46) possesses one (and only one) of the two following properties: 1) When $\lambda_1(A) \leq 0$ ($\lambda_1(A) > 0$), the optimal solution x^* of problem (2.4) (problem (2.2)) is the same as the optimal solution of problem (2.46) if and only if the hard case 2(ii) does not hold for problem (2.4) (problem (2.2)); 2) the optimal solution of problem (2.46) is $A_r^{\dagger}a$ ($A^{-1}a$) and lies in the interior of the unit ball if and only if the hard case 2(ii) holds for problem (2.4) (problem (2.2)). However, in our algorithm, we do not force $m_1 \geq \lambda_s(B)$ as this is not necessary for our needs. Mainly, choosing $\beta_j > 0$,

$j = 1, \ldots, i$ is enough for (2.46) to possess the desired property 1) or 2).

Note when the hard case 1 or 2(i) occurs and a solution lies on the boundary of the unit ball, the previous shifting procedure is still useful, since once we have done (2.44), $(-\infty, \hat{\gamma}_{\text{new}}] \subset (-\infty, \hat{\gamma}]$. Hence, the interval on the easy side where lies the desired γ_e of Section §2.5.1 is enlarged.

As a final remark, it should be understood that Section §2.5.1 provides a reliable way of treating the hard case and that what is additionally propose in this section is used to accelerate convergence in the hard case. We do not pretend that it is required to compute a basis of eigenvectors for the eigenvalue $\lambda_1(A)$ to handle the hard case, especially if i is large. Whenever an eigenvector v for $\lambda_1(B(\gamma))$ is computed and that $|a^T v|$ fall below some tolerance, we consider this as an indicator of the hard case and we reset B to $B + \beta v v^T$, where we chose $\beta = \max_{1 \leq i \leq n} B_{ii}$ as it is done in the Rendl-Wolkowicz Algorithm [10]. If the multiplicity of $\lambda_1(A)$ is large, this is unlikely to enhance the performance of the algorithm in the hard case (since we need i shifts, i.e. at least i iterations, to modify f) and we rely on taking steps to the boundary from hard side points to converge to an optimal solution. However, if i is small and through the iterations we are able to compute a basis of eigenvectors for $\lambda_1(A)$, then we are able to modify f in order to speed up our search of an approximate solution.

2.6 Further Implementation Issues and the Algorithm

Before stating our algorithm, we still need to discuss precisely how we intend to maximize f and update the information at a newly chosen value of γ, which we denote by γ_{new}. We denote the upper and lower bounds on γ^* by γ_U and γ_L. Analogously, let the bounds on r_G^2 be r_U^2 and r_L^2. Recall from Section §2.5.1 we denote by γ_e and γ_h values of γ which lie respectively on the easy and hard side.

2.6.1 Choosing $\bar{\lambda}$

We first need to specify a choice of $\bar{\lambda}$ which satisfies the inequalities (2.7). First, suppose first A has been reset to A_r. For $\bar{\lambda}$ to satisfy (2.7a), we need $\bar{\lambda} < 0$. Assume this holds. We have, using $B = Q^T(D - \bar{\lambda}I)Q$,

$$\|B^{-1}a\|^2 = \sum_{j=1}^{n} \frac{(q_j^T a)^2}{(\lambda_i(A_r) - \bar{\lambda})^2} \leq \frac{1}{(\lambda_1(A_r) - \bar{\lambda})^2} \sum_{j=1}^{n} (q_j^T a)^2 = \frac{1}{\bar{\lambda}^2}\|a\|^2.$$

Thus if we choose $\bar{\lambda} = -\|a\|$, then the inequalities (2.7) are satisfied. Second, suppose $\lambda_1(A) > 0$ and problem (2.2) is being solved. Then a similar analysis shows $\bar{\lambda} = \lambda_1(A) - \|a\|$ will satisfy as well the inequalities (2.7). Note that in both cases $B = A - (\lambda_1(A) - \|a\|)I$. Thus problem (2.13) is the same no matter if A is reset to A_r or not.

2.6.2 Initializing the Bounds

We now derive initial bounds on r_G^2 and γ^*. Trivially, we obtain

$$0 \leq r_G^2 = \gamma^* \lambda_1(B(\gamma^*)) \leq 1 \cdot \lambda_1(B) = \|a\|. \tag{2.49}$$

Hence, we let $r_L^2 = 0$ and $r_U^2 = \|a\|$. We have already obtained in Section §2.2 bounds on γ^*, namely $0 \leq \gamma^* \leq 1 - \|B^{-1}a\|^2$. We initially let $\gamma_U = 1 - \|B^{-1}a\|^2$, but a better lower bound on γ^* may be obtained using the optimality conditions for the semidefinite program (2.11).

Define

$$C := \begin{pmatrix} I_n & B^{-1}a & 0 \\ a^T B^{-1} & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_1 := \begin{pmatrix} 0 & 0 & I_n \\ 0 & 0 & 0 \\ I_n & 0 & 0 \end{pmatrix}, A_2 := \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & B \end{pmatrix}.$$

Hence, problem (2.11) may be rewritten as $\max\{r : C + rA_1 + \gamma A_2 \succeq 0\}$ and its dual as

$$\min\{\mathrm{tr}\,(C\Omega) : \mathrm{tr}\,(A_1\Omega) = -1, \mathrm{tr}\,(A_2\Omega) = 0, \Omega \succeq 0\}. \tag{2.50}$$

Note it is easy to show both semidefinite programs (2.11) and (2.50) are strictly feasible. Therefore, both semidefinite programs are solvable and there is a zero duality gap. Moreover, since the variables r and γ which satisfy the constraint of problem (2.11) are bounded, it may be approximately solved in polynomial time with a path following interior point method. A necessary and sufficient optimality conditions for (2.11) and (2.50) is the complementary slackness equation

$$\Omega^*(C + r_G A_1 + \gamma^* A_2) = 0, \tag{2.51}$$

where Ω^* is optimal for (2.50). Let

$$\Omega^* := \begin{pmatrix} M^* & m^* & N^* \\ m^{*^T} & p^* & g^{*^T} \\ N^* & g^* & T^* \end{pmatrix}, \tag{2.52}$$

where M^*, N^* and T^* are $n \times n$ matrices, m^* and g^* are $n \times 1$ vectors and p^* is a scalar. Feasibility of Ω^* yields

$$\mathrm{tr}\,(N^*) = -1/2 \quad \text{and} \quad \mathrm{tr}\,(T^*B) = p^*. \tag{2.53}$$

Optimality of Ω^* and (r_G, γ^*) yields

$$r_G = \mathrm{tr}\,(C\Omega^*) = \mathrm{tr}\,(M^*) + 2m^{*^T}B^{-1}a + p^*. \tag{2.54}$$

From (2.51), it may be deduced

$$r_G \mathrm{tr}\,(N^*) + \gamma^* \mathrm{tr}\,(T^*B) = 0, \tag{2.55a}$$

$$m^{*^T}B^{-1}a + p^*(1 - \gamma^*) = 0. \tag{2.55b}$$

Equations (2.53) and (2.55a) yield

$$p^* = \frac{r_G}{2\gamma^*}; \tag{2.56}$$

equations (2.55b) and (2.56) yield

$$m^{*^T} B^{-1} a = \frac{r_G(\gamma^* - 1)}{2\gamma^*}; \tag{2.57}$$

and equations (2.54), (2.56) and (2.57) yield

$$\mathrm{tr}\,(M^*) = \frac{r_G}{2\gamma^*}. \tag{2.58}$$

Since $\Omega^* \succeq 0$, the top left matrix of dimension $n+1$ in (2.52) is positive semidefinite, and applying Lemma 2.2, with $N = p^*$ (note $p^* > 0$), gives $M^* - \frac{1}{p^*} m^* m^{*^T} \succeq 0$. Taking the trace on both sides and using (2.56) and (2.58), we obtain

$$\|m^*\| \leq \frac{r_G}{2\gamma^*}. \tag{2.59}$$

From Equations (2.57) and (2.59), and the Cauchy-Schwartz inequality

$$\frac{(1-\gamma^*)r_G}{2\gamma^*} \leq \|m^*\| \|B^{-1}a\| \leq \frac{r_G}{2\gamma^*} \|B^{-1}a\|. \tag{2.60}$$

These inequalities give $1 - \|B^{-1}a\| \leq \gamma^*$. We thus set $\gamma_L = 1 - \|B^{-1}a\|$.

2.6.3 Updating the bounds

We now discuss how we may update the bounds on γ^* and r_G^2 assuming the values of f and its derivative f' are available at γ_e and γ_h (in our algorithm, these values of γ are respectively equal to γ_L and γ_U). The techniques we used are exactly those used in [10, 39] with the function $k(t)$ and we keep the same terminology.

To update the bounds on γ^*, we use a technique called *vertical cut*. Assume $f(\gamma_e) > f(\gamma_h)$. (A similar argument holds for the reverse inequality.) We find the intersection of the horizontal line through $(\gamma_e, f(\gamma_e))$ with the tangent line at the point $(\gamma_h, f(\gamma_h))$. Using the concavity of f, it is not hard to see we may update γ_U to:

$$\gamma_U \leftarrow \gamma_h + (f(\gamma_e) - f(\gamma_h))/f'(\gamma_h).$$

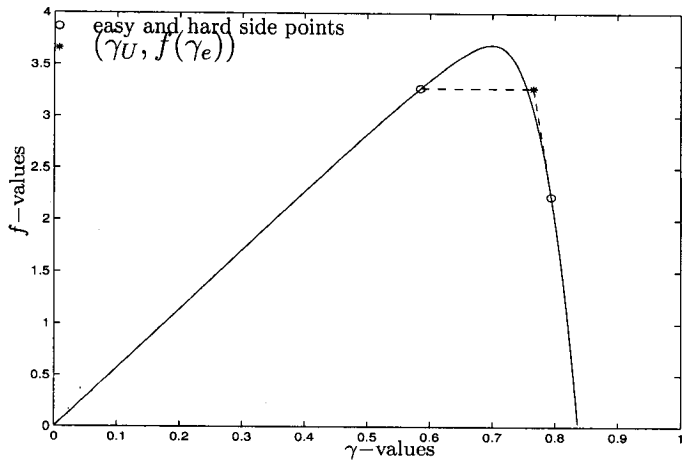

Figure 2.7: Vertical cut

Similarly, if the reverse inequality holds, we update γ_L. This technique is illustrated in Figure 2.7.

To update the bounds on r_G^2, we may trivially set $r_L^2 = \max\{f(\gamma_e), f(\gamma_h)\}$. An update on r_U^2 is obtained through *triangle interpolation*. Let $\bar{\gamma}$ be the γ coordinate of the point where the two tangent lines to f at γ_e and γ_h intersect. Then, from the concavity of f, we set $r_U^2 \leftarrow \min\{r_U^2, f(\gamma_h) + f'(\gamma_h)(\bar{\gamma} - \gamma_h)\}$. This is illustrated in Figure 2.8.

2.6.4 Generating a new iterate

At each iteration of our algorithm, we find a new iterate $\gamma_{\text{new}} \in (\gamma_L, \gamma_U)$ which is a better approximation to γ^*. As mentioned at the beginning of Section §2.5.1, unless the hard case 2(ii) occurs, problem (2.13) may be solved by finding the root to the equation $f'(\gamma) = 0$ (see Figure 2.9). Similarly to [10, 17, 32, 39], we find instead the root to the function

$$\psi(\gamma) := \lambda_1(B(\gamma))(1-\gamma)^2 - \frac{\gamma(a^T v)^2}{v^T B v}. \tag{2.61}$$

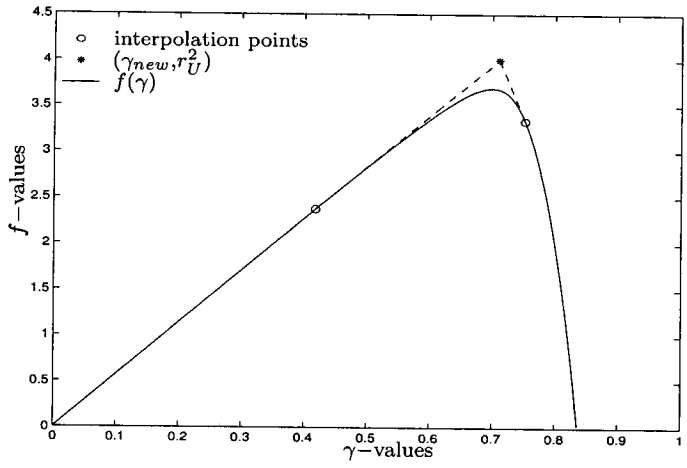

Figure 2.8: Triangle interpolation

This is clearly equivalent. This function has the advantage not to have an asymptote at $\gamma = 1$, thus it is in some sense less non-linear than $f'(\gamma)$ and interpolating on ψ will provide better estimates of γ^* (see Figure 2.10). Therefore, we will use, whenever possible, inverse linear (or quadratic) interpolation on $\psi(\gamma) = 0$ to obtain a better approximation to γ^*. Suppose we have computed the points (ψ_i, γ_i), $i = 1, 2, 3$. Then we solve the system

$$\begin{bmatrix} \psi_1^2 & \psi_1 & 1 \\ \psi_2^2 & \psi_2 & 1 \\ \psi_3^2 & \psi_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ \gamma_{\text{int}} \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}$$

and get the new estimate $\gamma_{\text{new}} = \gamma_{\text{int}}$, if $\gamma_{\text{int}} \in (\gamma_L, \gamma_U)$. We use the top 2×2 system in the linear interpolation case.

To generate a new iterate, we may also use triangle interpolation and set $\gamma_{\text{new}} = \bar{\gamma}$. This has the advantage to provide an update even in the hard case 2(ii).

2.6.5 The Algorithms

We are now ready to outline our main algorithm. We write the titles of the sections and subsections in capital letters. The algorithm solves (2.1), but focuses on solving

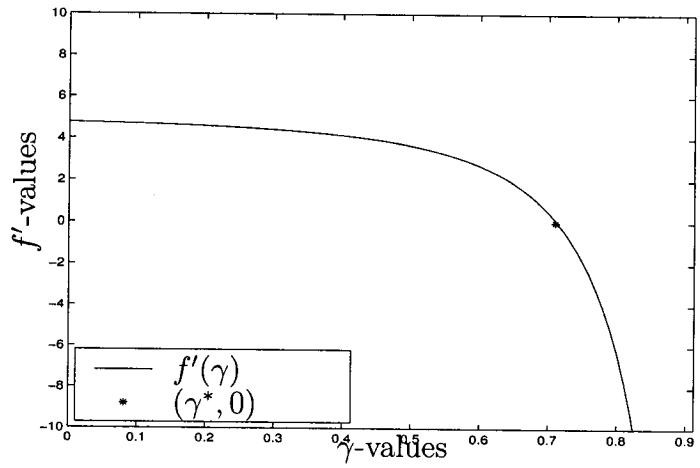

Figure 2.9: γ^* may be obtained by solving $f'(\gamma) = 0$.

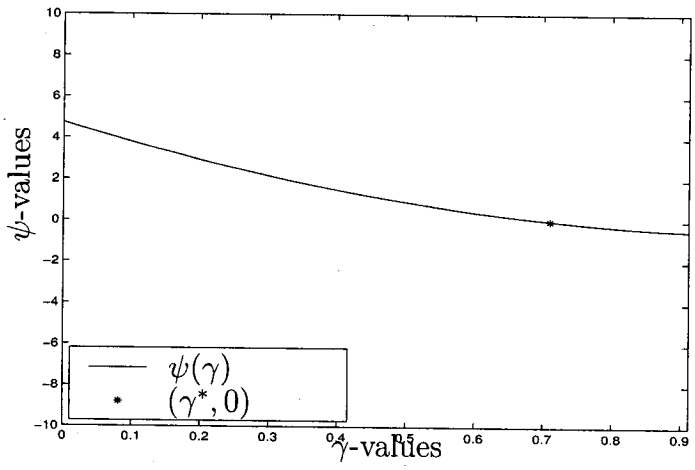

Figure 2.10: γ^* may be obtained by solving $\psi(\gamma) = 0$. The function has the advantage to be less nonlinear than $f'(\gamma)$.

(2.2) or (2.4). If $\lambda_1(A) \leq 0$ we set $\mu = \lambda_1(A)$ and reset A to A_r; otherwise, $\mu = 0$ and A is left unchanged. In either case, this is equivalent to resetting A to $A - \mu I$.

We rely on the theory in Section §2.4.2 if an unconstrained minimizer exists for problem (2.1) or the shifted problem (2.46). In the affirmative case, our algorithm halts and returns a solution to (2.1) using a conjugate gradient method and stepping to the boundary using (2.48) if needed when the hard case 2(ii) holds, the shift (2.44)

has been done and (2.46) is being solved.

Our algorithm has mainly four input parameters ϵ_k , $k = 1, 2, 3, 4$. The first two parameters are respectively the tolerance we allow on the relative duality gap and on the length of the interval of uncertainty $[\gamma_L, \gamma_U]$ for γ^*. We stop whenever, one of these two quantities falls below the tolerance. The third parameter is used as an indicator of the hard case. Specifically, if $v(\gamma)$ solves (2.14), then we use the appropriate techniques to handle the hard case whenever $|v(\gamma)^T a| < \epsilon_3$. The fourth parameter is used to check if a is nearly zero. The output returned is γ^*, x^* and λ^*, where the latter quantity is the Lagrange multiplier for (2.1). Theorem 2.3 suggests we should set $\lambda^* = \lambda_1(B(\gamma^*)) + \bar{\lambda} + \mu$. This is ideal if we return $x^* = x(\gamma)$ for γ close to the optimal γ^*. However, x^* may be returned after having taken a step to the boundary, i.e. x^* is nearly optimal, but the current value of γ may not be. Note the stationarity equation $(A - (\lambda^* + \mu)I)x^* = a$ and $\|x^*\| = 1$ imply $\lambda^* = x^{*T}Ax^* - a^T x^* + \mu$. We prefer the latter formula for λ^*, since it is more suitable if we want (x^*, λ^*) to satisfy stationarity.

Algorithm 2.6.1.

1. INITIALIZATION

   1.1. Compute $\lambda_1(A)$ and a corresponding unit eigenvector q_1.

   1.2 If $\|a\| \leq \epsilon_4$ and $\lambda_1(A) \geq 0$, exit and return $x^* = 0$, $\lambda^* = 0$.

   1.3 If $\|a\| \leq \epsilon_4$ and $\lambda_1(A) < 0$, exit and return $x^* = q_1$, $\lambda^* = \lambda_1(A)$.

   1.4. If $\lambda_1(A) > 0$, let $\mu = 0$, $\bar{\lambda} = \lambda_1(A) - \|a\|$, $B = A - \bar{\lambda}I$ and $x = q_1$.

   1.5. If $\lambda_1(A) \leq 0$, let $\mu = \lambda_1(A)$ and $\bar{\lambda} = -\|a\|$. Reset $A \leftarrow A - \lambda_1(A)I$ and let $B = A - \bar{\lambda}I$.

   1.6. Set $\gamma_L = 1 - \|B^{-1}a\|$, $\gamma_U = 1 - \|B^{-1}a\|^2$, $r_L^2 = 0$ and $r_U^2 = \|a\|$.

   1.7. If $|q_1^T a| < \epsilon_3$, do $\beta = \max_{1 \leq i \leq n} B_{ii}$, $B \leftarrow B + \beta q_1 q_1^T$, $\mathfrak{s} = 1$ and $r_U^2 = \beta + \|a\|$. Else, do $\mathfrak{s} = 0$ ($\mathfrak{s}$ indicates if a shift has been done).

2. ITERATION: While $\frac{r_U^2 - r_L^2}{1 + |q(x)|} > \epsilon_1$ and $\gamma_U - \gamma_L > \epsilon_2$.

2.1. FIND A BETTER APPROXIMATION TO γ^* AND UPDATE THE BOUNDS.

2.1.1. Do triangle interpolation if x_e and x_h have been defined (see §2.6.3 and §2.6.4); update r_U^2 and obtain γ_{new}.

2.1.2. Do vertical cut if x_e and x_h have been defined (see §2.6.3): update γ_L or γ_U.

2.1.3. Do linear or quadratic inverse interpolation on $\psi(\gamma)$ if at least two points are available (see §2.6.4): obtain γ_{new}.

2.1.4. If $\gamma_{\text{new}} \notin (\gamma_L, \gamma_U)$, set $\gamma = (\gamma_L + \gamma_U)/2$. Else, set $\gamma = \gamma_{\text{new}}$.

2.2. UPDATE THE INFORMATION AT γ.

2.2.1. Compute $\lambda_1(B(\gamma))$, $f(\gamma)$, $f'(\gamma)$ and a corresponding generalized eigenvector v which satisfies (2.14).

2.2.2. If $f(\gamma) > r_L^2$, do $r_L^2 = f(\gamma)$.

2.2.3. If $f'(\gamma) < 0$ (γ is on the hard side), do:

2.2.3.1. Let $\gamma_h = \gamma$, $v_h = v$, $x_h = \frac{1-\gamma}{a^T v} Bv$ and $\gamma_U = \gamma$.

2.2.3.2. If $\lambda_1(B(\gamma)) + \bar{\lambda} + (1 - \mathfrak{s})\mu > 0$, exit the algorithm and return

$$(x^*, \lambda^*) = \begin{cases} (A^{-1}a, 0) & \text{if } \mathfrak{s} = 0, \\ ((B + \bar{\lambda}I)^{-1}a + \alpha q_1, \mu) & \text{if } \mathfrak{s} = 1 \text{ and } \mu < 0, \\ ((B + \bar{\lambda}I)^{-1}a, \mu) & \text{if } \mathfrak{s} = 1 \text{ and } \mu = 0, \end{cases},$$

where α is defined from (2.43), substituting x_h for $(B + \bar{\lambda}I)^{-1}a$.

2.2.3.3. If x_e has been defined and $|v_e^T a| \geq \epsilon_3$, use (2.41) to obtain x_{new}. Else , use (2.42) to obtain x_{new}.

2.2.3.4. If $q(x_{\text{new}}) < q(x)$, do $x = x_{\text{new}}$, $r_U^2 = r^2(x)$.

2.2.4. If $f'(\gamma) > 0$ (γ is on the easy side), do:

2.2.4.1. Let $\gamma_e = \gamma$, $v_e = v$, $\gamma_L = \gamma$.

2.2.4.2. If $|v^T a| < \epsilon_3$, do $\beta = \max_{1 \leq i \leq n} B_{ii}$, $B \leftarrow B + \beta vv^T$, $\mathfrak{s} = 1$ and $r_U^2 = \beta + \|a\|$. Else, do $x_e = \frac{1-\gamma}{a^T v} Bv$.

2.2.4.3. If x_h and x_e have been defined use (2.41) to obtain x_{new}. Else,

$x_{\text{new}} = \frac{x_e}{\|x_e\|}$.

2.2.4.4. If $q(x_{\text{new}}) < q(x)$, do $x = x_{\text{new}}$, $r_U^2 = r^2(x)$.

3. RETURN THE OUTPUTS: $\gamma^* = \gamma$, $x^* = x$ and $\lambda^* = x^{*T}Ax^* - a^Tx^* + \mu$.

We now consider a simplified version of the previous algorithm for which we are able to provide convergence results. We also change our stopping criteria and use the ones introduced by Moré and Sorensen [32]. Our analysis follows closely the convergence analysis they provided for their algorithm. In our simplified version of Algorithm 2.6.1, we do not reset A, we do not shift the eigenvalues of A and we do not consider the bounds on r_G^2. The initial parameters are σ_1 and σ_2 in $(0,1)$, and $p \in (0, 1/2)$.

Algorithm 2.6.2.

1. INITIALIZATION

1.1. Compute $\lambda_1(A)$ and a corresponding unit eigenvector q_1.

1.2 If $\|a\| = 0$ and $\lambda_1(A) \geq 0$, exit and return $x^* = 0$, $\lambda^* = 0$.

1.3 If $\|a\| = 0$ and $\lambda_1(A) < 0$, exit and return $x^* = q_1$, $\lambda^* = \lambda_1(A)$.

1.4. Let $\bar{\lambda} = \lambda_1(A) - \|a\|$ and $B = A - \bar{\lambda}I$.

1.5. Set $\gamma_L = 1 - \|B^{-1}a\|$, $\gamma_U = 1 - \|B^{-1}a\|^2$.

1.6 Let $\mathfrak{c} = 0$ (number of consecutive points from the easy and hard side).

2. ITERATION: Repeat this step until termination

2.1. FIND A BETTER APPROXIMATION TO γ^* AND UPDATE THE BOUNDS.

2.1.1. Do triangle interpolation if x_e and x_h have been defined (see §2.6.3 and §2.6.4); obtain γ_{new}.

2.1.2. Do vertical cut if x_e and x_h have been defined (see §2.6.3): update γ_L or γ_U.

2.1.3. Do linear or quadratic inverse interpolation on $\psi(\gamma)$ if at least two points are available (see §2.6.4): obtain γ_{new}.

2.1.4. If $\mathfrak{c} = 4$ and (γ_h is not defined or $\gamma_U < \gamma_h$), then $\gamma = \gamma_U$.

Else, if $\gamma_{\text{new}} \notin [\gamma_L + p(\gamma_U - \gamma_L), \gamma_U - p(\gamma_U - \gamma_L)]$, let $\gamma = (\gamma_L + \gamma_U)/2$, otherwise let $\gamma = \gamma_{\text{new}}$.

2.2. UPDATE THE INFORMATION AT γ.

2.2.1. Compute $\lambda_1(B(\gamma))$, $f(\gamma)$, $f'(\gamma)$ and a corresponding generalized eigenvector v which satisfies (2.14).

2.2.2. If $f'(\gamma) < 0$ (γ is on the hard side), do:

2.2.2.1. Let $\gamma_h = \gamma$, $v_h = v$, $x_h = \frac{1-\gamma}{a^T v} Bv$, $\gamma_U = \gamma$ and $\mathfrak{c} = \min\{-1, \mathfrak{c} - 1\}$.

2.2.2.2. If $\lambda_1(B(\gamma)) + \bar{\lambda} > 0$, exit the algorithm and return

$(x^*, \lambda^*) = (A^{-1}a, 0)$.

2.2.2.3 If $|1 - \|x_h\|| \leq \sigma_1$ and $\lambda_1(B(\gamma)) + \bar{\lambda} \leq 0$, exit and return

$(x^*, \lambda^*) = (x_h, \lambda_1(B(\gamma)) + \bar{\lambda})$.

2.2.2.4. Let $z = q_1$ and compute α from (2.43).

If $\alpha^2 z^T (B - \lambda_1(B(\gamma))I)z \leq \sigma_1(2 - \sigma_1)\max\{\sigma_2, x_h^T a - (\bar{\lambda} + \lambda_1(B(\gamma)))\}$,

$(x^*, \lambda^*) = (x_h + \alpha z, x^{*^T} A x^* - a^T x^*)$. Exit the algorithm.

2.2.2.5. If x_e has been defined and $v_e^T a \neq 0$, let $z = \frac{x_e - x_h}{\|x_e - x_h\|}$ and compute α from (2.40).

If $\alpha^2 z^T (B - \lambda_1(B(\gamma))I)z \leq \sigma_1(2 - \sigma_1)\max\{\sigma_2, x_h^T a - (\bar{\lambda} + \lambda_1(B(\gamma)))\}$,

$(x^*, \lambda^*) = (x_h + \alpha z, x^{*^T} A x^* - a^T x^*)$. Exit the algorithm.

2.2.3. If $f'(\gamma) > 0$ (γ is on the easy side), do:

2.2.3.1. Let $\gamma_e = \gamma$, $v_e = v$, $\gamma_L = \gamma$ and $\mathfrak{c} = \max\{1, \mathfrak{c} + 1\}$.

2.2.3.2. If $v^T a \neq 0$, $x_e = \frac{1-\gamma}{a^T v} Bv$.

2.2.3.3. If $|1 - \|x_e\|| \leq \sigma_1$ and $\lambda_1(B(\gamma)) + \bar{\lambda} \leq 0$,

$(x^*, \lambda^*) = (x_e, \lambda_1(B(\gamma)) + \bar{\lambda})$. Exit the algorithm

2.2.3.4. If x_h and x_e have been defined and $v_e^T a \neq 0$, let $z = \frac{x_e - x_h}{\|x_e - x_h\|}$ and compute α from (2.40).

If $\alpha^2 z^T (B - \lambda_1(B(\gamma_h))I)z \leq \sigma_1(2 - \sigma_1)\max\{\sigma_2, x_h^T a - (\bar{\lambda} + \lambda_1(B(\gamma_h)))\}$,

$(x^*, \lambda^*) = (x_h + \alpha z, x^{*^T} A x^* - a^T x^*)$. Exit the algorithm.

2.7 Convergence Results

For our convergence analysis, we shall need the following lemma proved by Moré and Sorensen [32, Lemma 3.4].

Lemma 2.10. *Let* $0 < \sigma < 1$ *be given and suppose that* $A - \lambda I \succ 0$, $(A - \lambda I)x = a$ *and* $\lambda \leq 0$. *Let* $z \in \mathbb{R}^n$ *satisfy*

$$\|x + z\| = 1, \quad \text{and} \quad z^T(A - \lambda I)z \leq \sigma(x^T(A - \lambda I)x - \lambda), \tag{2.62}$$

then

$$-q(x + z) \geq (1 - \sigma)(x^T(A - \lambda I)x - \lambda) \geq (1 - \sigma)|q^*|, \tag{2.63}$$

where $q(x) = x^T Ax - 2a^T x$ *and* q^* *is the optimal solution to problem* (2.1).

Proof. For any $z \in \mathbb{R}^n$,

$$-q(x + z) = x^T(A - \lambda I)x - z^T(A - \lambda I)z - \lambda(x + z)^T(x + z). \tag{2.64}$$

For any $z \in \mathbb{R}^n$ which satisfy (2.62),

$$-q(x + z) \geq (1 - \sigma)(x^T(A - \lambda I)x - \lambda).$$

Moreover, if $q^* = q(x+z^*)$, where $\|x+z^*\| \leq 1$, then equality (2.64) and $z^T(A-\lambda I)z \geq 0$ implies

$$|q^*| = -q(x + z^*) \leq x^T(A - \lambda I)x - \lambda.$$

The last two inequalities imply the result. □

If x and z satisfy the conditions of Lemma 2.10, then $x + z$ is optimal if $q^* = 0$. otherwise $q^* < 0$ and $x + z$ is nearly optimal, i.e.

$$\frac{q(x+z) - q^*}{|q^*|} \leq \sigma.$$

We now prove Algorithm 2.6.2 terminates in a finite number of iterations.

Lemma 2.11. *Algorithm 2.6.2 terminates in a finite number of iterations with a solution x^* which satisfies*

$$q(x^*) - q^* \leq \sigma_1(2 - \sigma_1)\max\{|q^*|, \sigma_2\} \tag{2.65a}$$

$$\|x^*\| \leq 1 + \sigma_1. \tag{2.65b}$$

Proof. Suppose the contrary, i.e. that the algorithm does not terminate. Let $\{\gamma_k\}_{k\in\mathbb{N}}$ be the sequence of γ-iterates and denote by γ_{L_k} and γ_{U_k} respectively the current lower and upper bounds on γ^* after k iterations, where γ^* is optimal for problem (2.13). Notice the step 2.1.4 imply $\lim_{k\to\infty} \gamma_{U_k} - \gamma_{L_k} = 0$ and that there exist a subsequence $\{\gamma_{k_j}\}$ of $\{\gamma_k\}$ on the hard side such that $\lim_{j\to\infty} \gamma_{k_j} = \gamma^*$. Such a subsequence exists, since the step 2.1.4 insures that if the current interval of uncertainty $[\gamma_{L_k}, \gamma_{U_k}]$ for γ^* does not contain an iterate γ_h from the hard side and that if the last four iterates were on the easy side, then the next iterate is forced to be on the hard side.

First, assume the unique solution to problem (2.1) lies in the interior of the unit ball, i.e. $A \succ 0$ and $\|A^{-1}a\| < 1$. Now recall, from Lemmas 2.3 and 2.4, that

$$\begin{aligned} \lim_{\gamma\to-\infty} \lambda_1(B(\gamma)) + \bar{\lambda} &= \lambda_1(B) + \bar{\lambda} = \lambda_1(A) > 0, \\ \lim_{\gamma\to 1^-} \lambda_1(B(\gamma)) + \bar{\lambda} &= -\infty, \end{aligned}$$

and that $\lambda_1(B(\gamma)) + \bar{\lambda}$ is a decreasing function on the interval $(-\infty, 1)$. Hence there exists a unique solution in the interval $(-\infty, 1)$ to

$$\lambda_1(B(\gamma)) + \bar{\lambda} = 0,$$

say $\bar{\gamma}$. Note also that $\|x(\bar{\gamma})\| = \|A^{-1}a\| < 1$ and thus, by Lemma 2.7, $\gamma^* < \bar{\gamma}$. Hence for $\gamma \in (\gamma^*, \bar{\gamma})$, $\lambda_1(B(\gamma)) + \bar{\lambda} > 0$ and $\|x(\gamma)\| < 1$. Thus for j large enough, $\|x(\gamma_j)\| < 1$ and $\lambda_1(B(\gamma_j)) + \bar{\lambda} > 0$ and the algorithm would terminate in 2.2.2.2 and return the exact minimizer of problem (2.1), a contradiction.

Second, assume a solution to problem (2.1) lies on the boundary of the unit ball, i.e. an optimal solution for problem (2.2) is optimal for problem (2.1). In particular this implies

$$\lambda_1(B(\gamma_{k_j})) + \bar{\lambda} \leq 0 \quad \forall j. \tag{2.66}$$

If the hard case 2(ii) does not occur for problem (2.2), then, according to Corollary 2.2, the right-hand derivative of f at γ^* satisfies $f'(\gamma^{*^+}) = 0$. Thus $\lim_{j\to\infty} f'(\gamma_{k_j}) = 0$ and using Lemma 2.7 we deduce

$$\lim_{j\to\infty} \|x(\gamma_{k_j})\| = 1. \tag{2.67}$$

Equations (2.66) and (2.67) imply that for j large enough the algorithm would terminate in 2.2.2.3, a contradiction.

If the hard case 2(ii) occurs for problem (2.2), then $\lim_{j\to\infty} \lambda_1(B(\gamma_{k_j})) = \lambda_1(B)$. Hence

$$\lim_{j\to\infty} q_1^T(B - \lambda_1(B(\gamma_{k_j}))I)q_1 = 0 \tag{2.68}$$

and for j large enough the algorithm would terminate in 2.2.2.4, a contradiction.

Thus we have proved that Algorithm 2.6.2 terminates in a finite number of iterations and returns the couple (x^*, λ^*), which satisfies either one of the three following criteria:

1. $|1 - \|x^*\|| \leq \sigma_1$, $(A - \lambda^* I)x^* = a$, $\lambda^* \leq 0$, $A - \lambda^* I \succ 0$,

2. $\|x^*\| \leq 1$, $Ax^* = a$, $\lambda^* = 0$, $A \succ 0$,

3. $x^* = x_h + \alpha z$, $\|x^*\| = 1$, $\lambda^* = x^{*^T} A x^* - a^T x^*$, $(A - \lambda_h I)x_h = a$, $\lambda_h \leq 0$, $A - \lambda_h I \succ 0$, $\alpha^2 z^T (B - \lambda_1(B(\gamma_h))I)z \leq \sigma_1(2 - \sigma_1)\max\{\sigma_2, x_h^T a - \lambda_h\}$, where $x_h = x(\gamma_h)$, $\lambda_h := \lambda_1(B(\gamma_h)) + \bar{\lambda}$ and γ_h is on the hard side.

We now show that x^* satisfies the inequalities (2.65). First, if the criteria of items 1 or 2 are satisfied, then $|1 - \|x^*\|| \leq \sigma_1$ implies

$$1 - \sigma_1 \leq \|x^*\|, \tag{2.69a}$$

$$1 + \sigma_1 \geq \|x^*\|. \tag{2.69b}$$

Note the last inequality is the same as (2.65b). Using (2.64) and (2.69a), we obtain

$$\begin{aligned} -q(x^*) = -q(x^* + 0) &= x^{*^T}(A - \lambda^* I)x^* - \lambda^* \|x^*\|^2, \\ &\geq x^{*^T}(A - \lambda^* I)x^* - \lambda^*(1 - \sigma_1)^2, \\ &\geq (1 - \sigma_1)^2 (x^{*^T}(A - \lambda^* I)x^* - \lambda^*). \end{aligned} \tag{2.70}$$

Now if the optimal solution to problem (2.1) is $x^* + z^*$, then again using equation (2.64) and $z^{*^T}(A - \lambda^* I)z^* \geq 0$, we obtain

$$\begin{aligned} |q^*| = -q(x^* + z^*) &= x^{*^T}(A - \lambda^* I)x^* - z^{*^T}(A - \lambda^* I)z^* - \lambda^* \\ &\leq x^{*^T}(A - \lambda^* I)x^* - \lambda^*. \end{aligned} \tag{2.71}$$

Combining the inequalities (2.70) and (2.71) yields

$$-q(x^*) \geq (1 - \sigma_1)^2 |q^*| = -q^* + 2\sigma_1 q^* - \sigma_1^2 q^*.$$

Hence

$$\begin{aligned} q(x^*) - q^* &\leq \sigma_1(2 - \sigma_1)|q^*|, \\ &\leq \sigma_1(2 - \sigma_1)\max\{|q^*|, \sigma_2\}. \end{aligned}$$

Finally note the last inequality is the same as (2.65a).

Second, if the criteria of item 3 are satisfied, then consider first the case where $x_h^T a - \lambda_h > \sigma_2$. Then the assumptions of Lemma 2.10 are satisfied when σ is replaced by $\sigma_1(2-\sigma_1)$ and thus the inequality (2.65a) holds.

Now suppose $x_h^T a - \lambda_h \leq \sigma_2$. If the optimal solution to problem (2.1) is $x_h + z^*$, then similarly to how the inequality (2.71) was obtained, we obtain

$$|q^*| \leq x_h^T(A - \lambda_h I)x_h - \lambda_h = x_h^T a - \lambda_h \leq \sigma_2. \tag{2.72}$$

Combining the last inequality with equation (2.64) we obtain

$$\begin{aligned} q(x_h + \alpha z) &= -x_h^T(A - \lambda_h)x_h + \lambda_h + \alpha^2 z^T(A - \lambda_h I)z, \\ &\leq q^* + \alpha^2 z^T(A - \lambda_h I)z, \\ &= q^* + \alpha^2 z^T(B - \lambda_1(B(\gamma_h))I)z, \\ &\leq q^* + \sigma_1\sigma_2(2 - \sigma_1) \end{aligned} \tag{2.73}$$

Now the inequalities (2.72) and (2.73) give

$$q(x^*) - q^* \leq \sigma_1\sigma_2(2-\sigma_1) = \sigma_1(2-\sigma_1)\max\{\sigma_2, |q^*|\}.$$

Hence equation (2.65b) is satisfied and notice equation (2.65a) is trivially satisfied. □

Notice the assumption $\sigma_2 > 0$ is important in the previous proof so that equations (2.68) implies the algorithm eventually terminates if the hard case 2(ii) occurs in step 2.2.2.4. However, we may relax that assumption as shown in the following corollary.

Corollary 2.4. *If $\sigma_2 = 0$, then Algorithm 2.6.2 terminates in a finite number of iterations with either the optimal solution or a nearly optimal solution x^* which satisfies*

$$\frac{q(x^*) - q^*}{|q^*|} \leq \sigma_1(2-\sigma_1) \tag{2.74a}$$

$$\|x^*\| \leq 1 + \sigma_1. \tag{2.74b}$$

Proof. First, if $q^* = 0$, then $a = 0$ and $\lambda_1(A) \geq 0$. Hence, the algorithm terminates in the step 1.2 with the exact minimizer.

Second, if $|q^*| > 0$, then notice, for γ_h on the hard side and $x_h = x(\gamma_h)$, that,using equation (2.72), we obtain $|q^*| \leq x_h^T a - (\bar{\lambda} + \lambda_1(B(\gamma_h))$. Hence, $\max\{\sigma_2, x_h^T a - (\bar{\lambda} + \lambda_1(B(\gamma_h))\} > |q^*|$ and thus, if the hard case 2(ii) occurs, equation (2.68) still forces the algorithm to terminate in step 2.2.2.4. In the other easy and hard cases, the proof is identical to the one of Lemma 2.11. □

Typically, under appropriate conditions, trust-region methods will have a limit point where the gradient of the function minimized is zero and thus where the first optimality condition is satisfied. One possible condition is that at each iteration, the approximate solution to the trust-region subproblem attains a reduction in the quadratic model proportional to the reduction attained by the Cauchy point (the minimizer of q within the trust-region in the direction of steepest descent). To satisfy the second-order optimality condition, one needs to make better use of the quadratic term q. One possibility is that the approximate solution attains a reduction proportional to the reduction attained by the exact minimizer of the trust-region subproblem. The next Lemma shows equations (2.74) imply this result.

Lemma 2.12. *If $\sigma_2 = 0$, then Algorithm 2.6.2 returns an approximate solution x^* which satisfies, for some $\beta_1 > 0$ and $\beta_2 > 0$,*

$$-q(x^*) \geq \beta_1 |q^*|, \tag{2.75a}$$

$$\|x^*\| \leq \beta_2. \tag{2.75b}$$

Proof. From Corollary 2.4, either x^* satisfies equations (2.74) or it is the exact solution. In the first case, since $-q(x^*) \geq (1-\sigma_1)^2|q^*|$, let $\beta_1 = (1-\sigma_1)^2$ and $\beta_2 = 1+\sigma_1$. In the second case, notice this choice for β_1 and β_2 is still valid. □

2.8 A Trust-Region Method for Unconstrained Optimization

This section is a summary of Section §4 in [32]. We are interested in solving the unconstrained optimization problem

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

where f is a twice continuously differentiable function. The following algorithm is a standard trust-region method for solving such problem.

Algorithm 2.8.1. Trust Region Method

1. Given x_j and Δ_j, calculate $\nabla f(x_j)$ and $\nabla^2 f(x_j)$. Stop if
$$\frac{\|\nabla f(x_j)\|}{1+|f(x_j)|} < \epsilon. \tag{2.76}$$
2. Find δ_j, which approximately solves the following TRS:
$$\begin{aligned} \delta_j \in \quad & \arg\min \quad q_j(\delta) := \nabla f(x_j)^T\delta + \tfrac{1}{2}\delta^T\nabla^2 f(x_j)\delta \\ & \text{s.t.} \qquad \|\delta\|^2 \le \Delta_j^2. \end{aligned} \tag{2.77}$$
3. Evaluate $r_j = \frac{f(x_j)-f(x_j+\delta_j)}{q_j(0)-q_j(\delta_j)}$.
4. (a) If $r_j > 0.95$, set $\Delta_{j+1} = 2\Delta_j$ and $x_{j+1} = x_j + \delta_j$.
   (b) If $0.01 \le r_j < 0.95$, set $\Delta_{j+1} = \Delta_j$ and $x_{j+1} = x_j + \delta_j$.
   (c) If $r_j < 0.01$, set $\Delta_{j+1} = 0.5\Delta_j$ and $x_{j+1} = x_j$.

If in step 2 of Algorithm 2.8.1 we use Algorithm 2.6.2 with $\sigma_2 = 0$ to obtain an approximation to δ_j, then it follows from Lemma 2.12 that the Theorems of Section §4 in [32] hold. We recall their results and the reader is referred to this paper for the corresponding proofs.

The first Theorem says that if the level set of f at x_0 is bounded, then a limit point of the sequence $\{x_j\}$ will satisfy first and second order optimality conditions.

Theorem 2.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable on the level set $\Omega = \{x : f(x) \leq f(x_0)\}$ and consider the sequence $\{x_j\}$ produced by Algorithm 2.8.1, where δ_j in step 2 is obtained using Algorithm 2.6.2 to solve the TRS (2.77) with $\sigma_1 \in (0,1)$ and $\sigma_2 = 0$. If Ω is a compact set then either the algorithm terminates at x_l because $\nabla f(x_l) = 0$ and $\nabla^2 f(x_l) \succeq 0$, or $\{x_j\}$ has a limit point $x^* \in \Omega$ with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$.*

The second theorem implies that if the limit point of the previous theorem satisfies the sufficient second order optimality conditions, then the hole sequence $\{x_j\}$ converges to this limit point, eventually Algorithm 2.8.1 becomes Newton's method and the quadratic model at each iterate is a reliable approximation of the function f. The standard results of Newton's method on the rate of convergence thus apply.

Theorem 2.5. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable on the level set $\Omega = \{x : f(x) \leq f(x_0)\}$ and consider the sequence $\{x_j\}$ produced by Algorithm 2.8.1, where δ_j in step 2 is obtained using Algorithm 2.6.2 to solve the TRS (2.77) with $\sigma_1 \in (0,1)$ and $\sigma_2 = 0$. If x^* is a limit point of $\{x_j\}$ and $\nabla^2 f(x^*)$ is nonsingular, then $\{x_j\}$ converges to x^*, $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$. Furthermore, the bound $\|\delta\| \leq \Delta_j$ is inactive for sufficiently large j, $r_j \to 1$, and $\{x_j\}$ converges to x^* at a Q-superlinear rate of convergence. In addition, if $\nabla^2 f$ is Lipschitz continuous then the rate of convergence is Q-quadratic.*

2.9 A Trust-Region Method for Constrained Optimization

Having considered a trust-region method for unconstrained optimization, we now look at a trust-region method for solving an optimization problem with (possibly) nonlinear objective, (possibly) nonlinear inequality constraints and linear equality

constraints. Thus, we want to solve

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Ax = b \\ & c(x) \geq 0, \end{aligned} \tag{2.78}$$

where $x \in \mathbb{R}^n$, A is an $m \times n$ matrix with full row rank, $b \in \mathbb{R}^m$ and $c(x) \in \mathbb{R}^p$. We assume that $f(\cdot)$ and $c(\cdot)$ are twice contiguously differentiable and that there exists x_0 such that $Ax_0 = b$ and $c(x_0) > 0$. For ease of notation, if v is a vector, then V is the diagonal matrix obtained from v. We denote by e the vector of all ones and $J_{k,j}$ is the Jacobian matrix for the inequality constraints, i.e.

$$J_{k,j} = \begin{bmatrix} \nabla c_1(x_{k,j})^T \\ \dots \\ \nabla c_p(x_{k,j})^T \end{bmatrix}. \tag{2.79}$$

The algorithm we are about to present appears in Conn and al. [7] and is a primal-dual trust-region method. The method is based on lifting the inequality constraints in the objective through a log-barrier and aims at solving for $\mu_k \to 0$ the sequence of problems

$$\begin{aligned} \min \quad & \phi(x, \mu_k) := f(x) - \mu_k \textstyle\sum_{i=1}^{p} \log(c_i(x)) \\ \text{s.t.} \quad & Ax = b. \end{aligned} \tag{2.80}$$

It is primal-dual since the method iterates on the primal variable x and on a dual variable z for the inequality constraints of problem (2.78). In addition to the log-barrier, at each x-iterate, a trust-region forces the new iterate to stay in the set $\{x : c(x) > 0\}$. Since problem (2.80) is also solved iteratively, we shall denote by $x_{k,j}$ the x-iterate at the j-th iteration while solving this problem. A similar notation is used for other variables and functions. Thus the index k refers to the outer iteration and j to the inner iteration.

A key idea is to replace the function $\phi(x, \mu_k)$ by an appropriate quadratic model, the goal being to solve problem (2.80) through a sequence of trust-region subproblem

of the type (2.1). If $\phi(x, \mu_k)$ is approximated by a quadratic model obtained from the first three terms of a Taylor series expanded around $x_{k,j}$, we get

$$\begin{aligned}\phi(x_{k,j} + \Delta x_{k,j}, \mu_k) \approx\ & \phi(x_{k,j}, \mu_k) + \langle \nabla f(x_{k,j}), \Delta x_{k,j} \rangle + 1/2 \langle \Delta x_{k,j}, \nabla^2 f(x_{k,j}) \Delta x_{k,j} \rangle \\ & -\mu_k \langle J_{k,j}^T C_{k,j}^{-1} e, \Delta x_{k,j} \rangle + 1/2 \mu_k \langle \Delta x_{k,j}, J_{k,j}^T C_{k,j}^{-2} J_{k,j} \Delta x_{k,j} \rangle \\ & -1/2 \mu_k \sum_{i=1}^{p} \frac{1}{c_i(x_{k,j})} \langle \Delta x_{k,j}, \nabla^2 c_i(x_{k,j}) \Delta x_{k,j} \rangle. \end{aligned} \tag{2.81}$$

However, according to the comments in [7], this quadratic approximation does not model the log-barrier function very well near the boundary of $\{x : c(x) \geq 0\}$. This has the effect in practice of slowing down convergence. Thus, the second order term in (2.81) is replaced by a term whose growth is less dominant. Namely, the matrix

$$\nabla^2 f(x_{k,j}) + \mu_k J_{k,j}^T C_{k,j}^{-2} J_{k,j} - \sum_{i=1}^{p} \frac{\mu_k}{c_i(x_{k,j})} \nabla^2 c_i(x_{k,j})$$

is replaced by

$$W_{k,j} := \nabla^2 f(x_{k,j}) + J_{k,j}^T C_{k,j}^{-1} Z_{k,j} J_{k,j} - \sum_{i=1}^{p} (z_k)_i \nabla^2 c_i(x_{k,j}), \tag{2.82}$$

where $Z_{k,j}$ is a bounded positive diagonal matrix. Thus the quadratic model of $\phi(x, \mu_k)$ at $x_{k,j}$ is

$$\begin{aligned} m_{k,j}(x_{k,j} + \Delta x_{k,j}, \mu_k) :=\ & \phi(x_{k,j}, \mu_k) + \langle \nabla f(x_{k,j}) - \mu_k J_{k,j}^T C_{k,j}^{-1} e, \Delta x_{k,j} \rangle \\ & +1/2 \langle \Delta x_{k,j}, W_{k,j} \Delta x_{k,j} \rangle. \end{aligned} \tag{2.83}$$

Assuming $Ax_{k,0} = b$, then at the j-th iteration, we may obtain $x_{k,j+1}$ by solving the quadratic program

$$\begin{aligned} \min_{\Delta x_{k,j}} \quad & m_{k,j}(x_{k,j} + \Delta x_{k,j}, \mu_k) \\ \text{s.t.} \quad & A \Delta x_{k,j} = 0. \end{aligned} \tag{2.84}$$

However, we have no guarantee the latter problem has a solution. This justifies the addition of a trust-region constraint. This constraint will also be of use in the

algorithm for enforcing $c(x_{k,j} + \Delta x_{k,j}) > 0$. Thus, at each inner iteration of the algorithm, the trust-region subproblem

$$\begin{aligned} \min_{\Delta x_{k,j}} \quad & m_{k,j}(x_{k,j} + \Delta x_{k,j}, \mu_k) \\ \text{s.t.} \quad & A\Delta x_{k,j} = 0 \\ & \|\Delta x_{k,j}\| \leq \Delta_{k,j}, \end{aligned} \tag{2.85}$$

is solved. Note if N is a basis for the null space of A, then the change of variable $\Delta x_{k,j} = N s_{k,j}$ transforms the trust-region subproblem (2.85) exactly in the form (2.1). Precisely, (2.85) may be solved if we can solve the TRS

$$\begin{aligned} \min_{s_{k,j}} \quad & \langle N^T(\nabla f(x_{k,j}) - \mu_k J_{k,j}^T C_{k,j}^{-1} e), s_{k,j} \rangle + 1/2 \langle s_{k,j}, N^T W_{k,j} N s_{k,j} \rangle \\ \text{s.t.} \quad & \|s_{k,j}\| \leq \Delta_{k,j}, \end{aligned} \tag{2.86}$$

Before stating the algorithm, we consider another motivation for using the quadratic model $m_{k,j}(x_{k,j}, \mu_k)$. Consider the first order optimality conditions for problem (2.78), namely

$$\nabla f(x) + A^T y - J(x)^T z = 0, \quad Ax = b, \quad C(x)z = 0 \quad c(x) \geq 0, \quad z \geq 0,$$

where z and y are Lagrange multipliers. Now, if we perturb the complementary slackness equation on the right hand side, introducing a perturbation $\mu e > 0$, we obtain

$$\nabla f(x) + A^T y - J(x)^T z = 0, \quad Ax = b, \quad C(x)z = \mu e \quad c(x) \geq 0, \quad z \geq 0.$$

Let $y_{k,j+1} = y_{k,j} + \Delta y_{k,j}$. Applying Newton's method to the previous system of equations at some iterate $(x_{k,j}, z_{k,j}, y_{k,j})$ yields the system

$$\begin{pmatrix} W_{k,j} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x_{k,j} \\ y_{k,j+1} \end{pmatrix} = - \begin{pmatrix} \nabla f(x_{k,j}) - \mu_k J_{k,j}^T C_{k,j}^{-1} e \\ 0 \end{pmatrix}, \tag{2.87a}$$

$$\Delta z_{k,j} = -z_{k,j} - C_{k,j}^{-1} Z_{k,j} J_{k,j} \Delta x_{k,j} + \mu_k C_{k,j}^{-1} e. \tag{2.87b}$$

Now notice the system (2.87a) gives the first-order optimality conditions for problem (2.84). This gives another motivation for replacing the second order term in (2.81) by $W_{k,j}$. Equation (2.87b) will also be used to update $z_{k,j}$, but a proper safeguarding procedure is used to guarantee $z_{k,j} + \Delta z_{k,j} > 0$. We now outline the algorithm which is split for convenience into its inner and outer iterations. We use $\mathcal{P}_{\mathcal{I}}[v]$ to denote the component-wise projection of the vector v onto the interval $\mathcal{I}$.

Algorithm 2.9.1.

1. INITIALIZATION: An initial point x_0 that satisfies $Ax_0 = b$ and $c(x_0) > 0$ is given. Let $z_0 = \mu_0 C(x_0)^{-1}e$. Set $k = 0$ and $\mu_0 = 10$.
2. ITERATION: While $\mu_k >$ tol or $k = 0$
   2.1. Let $\zeta_k = 10^{-k}$ and minimize the log-barrier function $\phi(x, \mu_k)$ starting from $(x_k, z_k) \equiv (x_{k,0}, z_{k,0})$ using Algorithm 2.9.2 and obtain (x_{k+1}, z_{k+1})
   2.2. $\mu_{k+1} = \min(0.1\mu_k, \mu_k^{1.5})$ and $k = k + 1$.

Algorithm 2.9.2.

1. INITIALIZATION: An initial point $x_{k,0}$ that satisfies $Ax_{k,0} = b$ and $c(x_{k,0}) > 0$, a vector of dual variables $z_{k,0} > 0$ and $\zeta_k \in (0.1)$ are given. Set $j = 0$ and let $\Delta_{k,0} = 10\mu_k$.
2. ITERATION: while $\|C_{k,j}z_{k,j} - \mu_k e\|_\infty > \mu_k^{1.01}$, $\|N^T(\nabla f(x_{k,j}) - J_{k,j}^T z_{k,j})\| > \mu_k^{1.01}$ and $\lambda_1(N^T W_{k,j} N) > -\mu_k^{1.01}$
   2.1. Approximately solve the trust-region subproblem (2.85).
   2.2. If $c(x_{k,j} + \Delta x_{k,j}) \geq \zeta_k c(x_{k,j})$, let $\rho_{k,j} = \frac{\phi(x_{k,j},\mu_k) - \phi(x_{k,j}+\Delta x_{k,j},\mu_k)}{m_{k,j}(x_{k,j},\mu_k) - m_{k,j}(x_{k,j}+\Delta x_{k,j},\mu_k)}$. Else $\rho_{k,j} = -\infty$.
   2.3. If $\rho_{k,j} \geq 0.01$, $x_{k,j+1} = x_{k,j} + \Delta x_{k,j}$. Else $x_{k,j+1} = x_{k,j}$.
   2.4. Set $\Delta_{k,j+1} = \begin{cases} \min(10^{20}, \max(2\|N^T \Delta x_{k,j}\|, \Delta_{k,j})), & \text{if } \rho_{k,j} \geq 0.9, \\ \Delta_{k,j} & \text{if } \rho_{k,j} \in [0.01, 0.9), \\ 1/2\Delta_{k,j} & \text{if } \rho_{k,j} < 0.01. \end{cases}$

2.5. Let $\mathcal{I} = [1/2 \min(e, z_{k,j}, \mu_k C_{k,j}^{-1} e), \max(10^{20} e, z_{k,j}, 10^2 0 \mu_k^{-1} e, 10^{20} \mu_k C_{k,j}^{-1} e)]$.
Let $\Delta z_{k,j} = -z_{k,j} - C_{k,j}^{-1} Z_{k,j} J_{k,j} \Delta x_{k,j} + \mu_k C_{k,j}^{-1} e$.

$$\text{Let } z_{k,j+1} = \begin{cases} \mathcal{P}_{\mathcal{I}}[z_{k,j} + \Delta z_{k,j}] & \text{if } x_{k,j+1} = x_{k,j} + \Delta x_{k,j}, \\ z_{k,j} & \text{if } x_{k,j+1} = x_{k,j}. \end{cases}$$

2.6. $j = j + 1$.

Although stating the convergence result for Algorithm 2.9.1 would require a longer exposure of the original paper than we want to give here, one may expect that a finite limit point (x^*, z^*) will satisfy the optimality conditions for (2.78), namely

$$N^T(\nabla f(x^*) - J(x^*)^T z^*) = 0, \quad Ax^* = b, \quad C(x^*)z^* = 0 \quad c(x^*) \geq 0, \quad z^* \geq 0,$$
$$\langle s, (\nabla^2 f(x^*) - \sum_{i=1}^{p} z_i^* \nabla^2 c_i(x^*))s \rangle \geq 0, \quad \text{for all} \quad s \in \mathcal{U},$$

where

$$\mathcal{U} = \left\{ s \;\middle|\; \begin{array}{c} As = 0, \\ [J(x^*)s]_i = 0 \quad \text{if } c_i(x^*) = 0. \end{array} \right\}$$

2.10 Numerical Results

The goal of this section is to compare Algorithm 2.6.1 with other existing TRS algorithms, particularly the Rendl-Wolkowicz Algorithm, and to briefly give results obtained by using the trust-region algorithms 2.8.1 and 2.9.1 to respectively solve unconstrained and constrained optimization problems. Although we believe the interest in Algorithm 2.6.1 is mainly theoretical, since we do not expect it to be faster or more robust than other existing methods, our hope is that it can perform reasonably well. All algorithms were implemented using MATLAB 6.5 and computations were done on a Pentium 4 at 1.8GHz with 256MB of memory (all codes may be found at the following URL: www.math.mcgill.ca/~fortin). Unless stated otherwise, MATLAB's implementation of ARPACK, the function `eigs`, was chosen to compute required

eigenvalues and eigenvectors and we used $\min(m/2, 40)$ basis vectors before each implicit restart, where m is the size of the eigenvalue problem considered. Whenever Algorithm 2.6.1 is used, $\epsilon_2 = 10^{-13}$, $\epsilon_3 = \max(10^{-8}, \epsilon_1 10^{-6})$ and $\epsilon_4 = 10^{-13}$.

2.10.1 Comparing different TRS algorithms

Our first concern is to use different algorithms for solving trust-region subproblems (our test problems are similar to those used in [40]) and compare the performance of Algorithm 2.6.1. We should not expect Algorithm 2.6.1 to perform better than the Rendl-Wolkowicz Algorithm, since at each iteration we require `eigs` to compute the smallest eigenvalue of a non-symmetric parameterized matrix (i.e. the matrix $B - 1/(1-\gamma)B^{-1}aa^T$), where as the parameterized matrix $D(t)$ in the Rendl-Wolkowicz Algorithm is symmetric. It would also be possible to find instead the smallest eigenvalue of the symmetric generalized eigenvalue problem (2.14), but in the version of MATLAB used by the author, the function `eigs` did not return accurate eigenvalue estimates on such problems. Nevertheless, this should also lead to an algorithm at least as expensive as the Rendl-Wolkowicz Algorithm. However, our hope is that Algorithm 2.6.1 is not much more expensive and that there is more than just theoretical interest in the method.

The General Case

In the first experiment, trust-region subproblems were generated by choosing $A = 1/2(L - 5I)$, where L is the two-dimensional discrete Laplacian on the unit square based upon a 5-point stencil with equally mesh points. The shift $-5I$ makes A indefinite. The components of the vector a and the radius Δ are randomly generated and respectively uniformly distributed on the intervals $(-2, 0)$ and $(0, 100)$. We compared Algorithm 2.6.1, the Rendl-Wolkowicz Algorithm (RW), the Moré-Sorensen Algorithm (MS), the difference of convex functions algorithm of An and Tao [47]

(DCA) and the generalized Lanczos trust-region method of Gould, Lucidi, Roma and Toint [17] (GLTR). It should be noted that the GLTR and the DCA Algorithms are personal implementations in MATLAB by the author of this thesis. For each TRS generated, we first computed an approximate solution with the GLTR Algorithm and stopped when the iterates (x_k, λ_k) satisfied

$$\|(A - \lambda_k I)x_k + a\| < 1e - 8 \tag{2.88}$$

or 40 iterations beyond the Steihaug-Toint point have occurred (see [17] for the relevant terminology). For the last four algorithms, we stopped once a solution as accurate was obtained. In the DCA Algorithm, we have set $\rho = 0.25\|A\|_1$.

We report the computation time, the number of matrix-vector multiplications and the number of iterations (in case of the GLTR Algorithm we report the number of iterations beyond the Steihaug-Toint point) needed for each algorithm to converge. Since the MS Algorithm requires a Cholesky factorization at each iteration, the number of matrix-vector multiplications for this algorithm is irrelevant. The numbers indicate the average obtained by generating five different TRS for each problem size. Note this is also true for all numbers in the other tables that appear in this section. The results appear in Tables 2.3, 2.4 and 2.5.

As we expected, Algorithm 2.6.1 and the Rendl-Wolkowicz algorithm behave similarly, although the latter algorithm uses less matrix-vector multiplications and consequently its computation times are smaller. This is obviously due to the fact that the eigenvalue problems it has to solve are simpler. In terms of computation time, it appears clear that the Moré-Sorensen algorithm becomes outperformed by all other algorithms and we stopped testing the algorithm beyond the problem size 2500, because of the increasing time taken to compute the Cholesky factorizations. However, when we compare the four remaining algorithms, we were surprise to observe how quick the GLTR and DCA algorithms were and how few matrix-vector multiplications were needed to achieve the same accuracy obtained with the other algorithms.

Problem Size	TRS Algorithm				
	GLTR	Alg. 2.6.1	MS	RW	DCA
25	0.222	0.194	0.030	0.198	0.058
100	0.186	0.418	0.102	0.326	0.046
225	0.146	0.452	0.296	0.358	0.040
400	0.128	0.510	0.664	0.426	0.056
625	0.158	0.620	1.120	0.488	0.038
900	0.138	0.778	2.164	0.598	0.052
1225	0.126	1.050	3.852	0.802	0.080
2500	0.160	1.856	15.950	1.412	0.078
22500	0.512	24.474	-	16.500	0.360
62500	1.058	99.612	-	76.894	0.868
122500	2.104	262.320	-	207.542	1.770

Table 2.3: Computation times for different TRS algorithms on problems involving the 2D discrete Laplacian.

Notice the DCA Algorithm required only 5 iterations and matrix-vector multiplications for the larger problems. It remains to see if such behavior can be observed in the hard case 2ii).

The Hard Case 2ii)

In our second experiment, we wanted to test the performance of the TRS algorithms on hard case 2ii) TRS. We chose $A = 1/2UDU^T$ with D diagonal, $U = I - 2uu^T$ and $\|u\| = 1$. The elements of D were randomly generated with uniform distribution on $(-5, 5)$ then sorted in decreasing order and D_{jj} set to -5 for $j = 1, \ldots, s$, allowing multiplicity s for the smallest eigenvalue of A. The eigenvalues of A are D_{jj} with

Problem Size	TRS Algorithm			
	GLTR	Alg. 2.6.1	RW	DCA
25	57.0	130.8	117.0	35.2
100	57.0	334.2	240.0	23.2
225	49.0	328.0	240.0	17.2
400	45.8	342.0	260.0	14.6
625	45.0	338.6	260.0	12.4
900	41.0	355.0	280.0	11.0
1225	37.8	373.0	300.0	10.0
2500	37.0	397.8	320.0	9.0
22500	29.0	656.0	480.0	6
62500	25.0	933	784	5
122500	25.0	1210	1072	5

Table 2.4: Number of matrix-vector multiplications for different TRS algorithms on problems involving the 2D discrete Laplacian.

corresponding eigenvectors $q_j = e_j - 2uu_j$, where e_j is the j-th column of the identity matrix. The vectors u and g have entries uniformly distributed on the intervals $(-0.5, 0.5)$. The vector u is normalized to have unit length and density $\sqrt{5/n}$ so that the matrix A has density $5/n$, i.e. A has $5n$ nonzeros. If $s = 1$, the vector g is orthogonalized against q_1 and if $s > 1$ it is reset to $g \leftarrow \sum_{j=s+1}^{n} g_j q_j$. Then a noise vector w of norm 10^{-8} is generated and g is reset to $g \leftarrow (g + w)/\|g + w\|$. Finally $a = -1/2g$ and $\Delta = 2\|(A - \lambda_1(A)I)^{\dagger}a\|$ to force the hard case 2ii).

We first considered TRS where $s = 1$. Each TRS was first solved using Algorithm 2.6.1 with $\epsilon_1 = 10^{-12}$. Then, the other algorithms were halted when a solution with equal or smaller objective value was obtained, or when the objective value was within

Problem Size	TRS Algorithm				
	GLTR	Alg. 2.6.1	MS	RW	DCA
25	13.0	4.4	6.0	5.2	35.2
100	13.0	5.0	6.0	5.0	23.2
225	11.0	5.0	5.0	5.0	17.2
400	10.2	5.0	4.8	5.0	14.6
625	10.0	5.0	4.0	5.0	12.4
900	9.0	5.0	4.0	5.0	11.0
1225	8.2	5.0	4.0	5.0	10.0
2500	8.0	5.2	4.0	5.0	9.0
22500	6.0	6.0	-	3.0	6.0
62500	5.0	6.0	-	3.6	5.0
122500	5.0	6.0	-	3.8	5.0

Table 2.5: Number of iterations for different TRS algorithms on problems involving the 2D discrete Laplacian.

10^{-10} from the optimal objective value obtained with Algorithm 2.6.1. We compare Algorithm 2.6.1 to the Moré-Sorensen Algorithm and the Rendl-Wolkowicz Algorithm only, since the GLTR Algorithm is not suited to handle the hard case 2ii) and the DCA Algorithm in our experience, on our test problems, has very slow convergence in the hard case 2ii) (more than 5000 matrix-vector multiplications needed). However, for this set of problems, we were unable to use the function `eigs` in Algorithm 2.6.1 for dimensions higher than $n = 2500$, since the former function often failed to give the required eigenvalues and eigenvectors. This highlights the fact that our method is quite dependent on the robustness of the eigenvalue solver and explains why the larger size problems do not appear in our results. For all three algorithms considered, we

compare the computation time and the number of iterations needed to converge. We also report for Algorithm 2.6.1 and the RW Algorithm the number of matrix-vector multiplications and the number of shifts (the ones of Section §2.5.2) that are used. Results are given in Table 2.6.

	Computation time			Iterations			matrix-vector ×		Shifts	
Problem Size	Alg. 2.6.1	R-W	M-S	Alg. 2.6.1	R-W	M-S	Alg. 2.6.1	R-W	Alg. 2.6.1	R-W
25	0.3567	0.5733	0.0900	1.4	7.6	21.6	329.7	482.7	1.0	1.7
225	1.8933	4.0667	2.1900	3.8	7.6	24.0	1174.0	1941.3	1.0	4.7
625	20.2500	46.2733	12.2033	3.4	9.2	29.8	1890.0	3580.0	1.0	5.7
1225	138.2267	462.5367	46.8767	5.0	10.8	25.4	3130.0	9646.0	1.0	5.7

Table 2.6: Comparison of Algorithm 2.6.1, the Rendl-Wolkowicz Algorithm and the Moré-Sorensen Algorithm for hard case 2ii) TRS with a multiplicity of one for $\lambda_1(A)$.

Two things may be noted from Table 2.6. First, it is surprising that the Moré-Sorensen is faster than its two other competitors. One possible explanation is that one can show the Cholesky factors keep for this special set of problems the same density as the matrix A. In other words, sparsity is not lost through the Cholesky factorizations. Also, it appears that for Algorithm 2.6.1 and the Rendl-Wolkowicz Algorithm, many matrix-vectors multiplications are required for computing the eigenvalues. Indeed, when comparing with the results of Table 2.4, we see that as much as ten times more matrix-vectors multiplications were needed for similar size TRS. It thus seems the difficulty of the TRS created is reflected in the effort required for computing the eigenvalue and eigenvector at each iteration.

Second, we were surprised to find that Algorithm 2.6.1 did better than the Rendl-Wolkowicz Algorithm. This seems to be caused by the fact that at least twice as many iterations are needed by the former algorithm to converge. This can partly be

explained by the different heuristics used in each respective algorithms or by the fact that up to six shifts may be done by the Rendl-Wolkowicz algorithm even though the multiplicity of $\lambda_1(A)$ in these examples is only one.

Table 2.7 gives the result of a similar experience, but this time with $s = 5$, i.e. the smallest eigenvalue of the matrix A has multiplicity five. This time `eigs` failed to return accurate eigenvalues even for the smallest size problems. We thus turned to using `eig` and did not go beyond TRS of size $n = 625$.

	Computation time			Iterations			Shifts	
Problem Size	Alg. 2.6.1	R-W	M-S	Alg. 2.6.1	R-W	M-S	Alg. 2.6.1	R-W
25	0.176	0.588	0.098	7.4	5.6	19.8	5.0	2.6
225	1.028	2.006	0.504	7.8	7.6	23.6	5.0	3.2
625	8.352	4.862	1.834	7.4	8.6	25.6	5.0	3.4

Table 2.7: Comparison of Algorithm 2.6.1, the Rendl-Wolkowicz Algorithm and the Moré-Sorensen Algorithm for hard case 2ii) TRS with a multiplicity of five for $\lambda_1(A)$.

This time, the Rendl-Wolkowicz Algorithm did less than five shifts before it converged and it is not as clear as in Table 2.6 if Algorithm 2.6.1 performed better.

As we have mentioned above, the example created have the advantage for the Moré-Sorensen Algorithm of preserving the sparsity of the matrix A in the Cholesky factorizations. We thus wanted to look at other examples where this would not happen. Hence we have created examples where the matrix A has its first $\log(n)$ rows and columns fully dense. This creates fully dense Cholesky factorizations in the Moré-Sorensen Algorithm. The small code used for generating our problems in given in Appendix A. Our problems have approximately $5n$ non-zeros, where n is the problem size and the multiplicity of $\lambda_1(A)$ is one. Computation times are illustrated in Figure 2.11 and Table 2.8 gives the number of iterations and matrix-vector multiplications

required. Note there was no problem in using `eigs` on this set of problems when computing eigenvalues and eigenvectors were required.

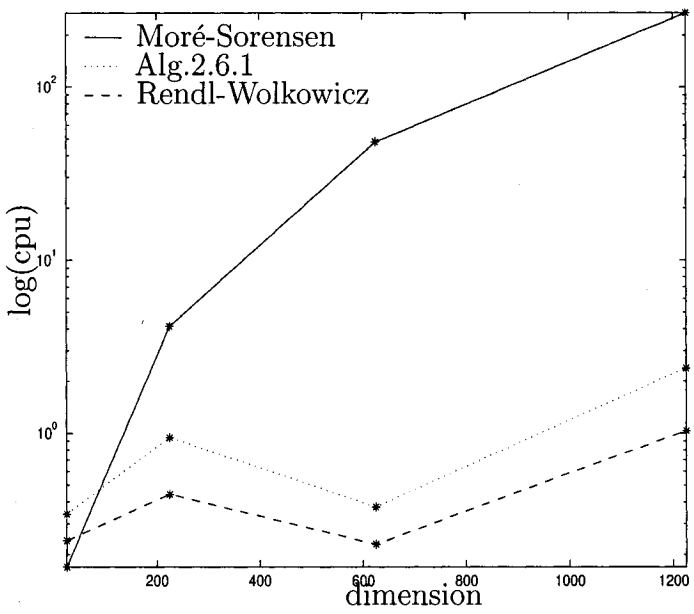

Figure 2.11: Logarithm of the computation time (seconds) in function of problem dimensions of three algorithms used to solve hard case 2ii) TRS with first rows and columns fully dense.

Figure 2.11 shows the consequence of a loss of sparsity for the Moré-Sorensen Algorithm: the computation times are much slower than those obtained by the other two algorithms. Few iterations are needed for convergence by these two latter algorithms and only one shift is used. In this case there is no doubt the Rendl-Wolkowicz performs better than Algorithm 2.6.1 due to a smaller number of iterations to solve each problem and consequently less matrix-vector multiplications are required.

2.10.2 Algorithm 2.8.1 for Unconstrained Optimization

Our goal here is to briefly study the behavior of Algorithm 2.8.1 in function of the TRS algorithms (all algorithms that appear in Section §2.10.1 except for the Moré-

	Iterations			Matrix-vector ×		Shifts	
Problem Size	Alg. 2.6.1	R-W	M-S	Alg. 2.6.1	R-W	Alg. 2.6.1	R-W
25	2.8	2.2	26.2	341.2	189.4	1.0	1.0
225	3.8	2.2	32.8	688.6	332.2	1.0	1.0
625	1.0	1.0	31.8	213.4	95.0	1.0	1.0
1225	2.6	1.6	30.0	655.2	336.2	1.0	1.0

Table 2.8: Number of iterations, matrix-vector multiplications and shifts for three algorithms used to solve hard case 2ii) TRS with the $\log(n)$ first rows and columns fully dense.

Sorensen Algorithm, since the problems are not of small size) used to approximately solve the TRS (2.77). Our test problems are taken from the CUTEr [18] package.

When Algorithm 2.6.1 and the Rendl-Wolkowicz Algorithms are used to solve problem (2.77), we stop when the relative duality gap is smaller than

$$\epsilon_1 := \max(10^{-12}, \min(10^{-6}, 10^{-3}\nabla f(x_j)).$$

When using the GLTR Algorithm, we consider the couple (δ_k, λ_k) of approximate solution and corresponding Lagrange multiplier a reasonable approximation if

$$\|(\nabla^2 f(x_j) - \lambda_k I)\delta_k + \nabla f(x_j)\| < \sqrt{\epsilon_1}$$

([10] shows the expression on the left-hand side is of the same order as the square root of a duality gap for problem (2.77)) or if 10 iterations were done beyond the Steihaug-Toint point. Finally when the DCA Algorithm was used, we chose in their algorithm to set $\epsilon = \epsilon_1$ and $\rho = 0.25\|A\|_1$. The results appear in Table 2.9 where we give the number of iterations taken by Algorithm 2.8.1 to satisfy

$$\frac{\nabla f(x_j)}{|f(x_j)| + 1} < 10^{-5}$$

		Iterations				Computation time			
Problem Name	Size	Alg. 2.6.1	R-W	GLTR	DCA	Alg. 2.6.1	R-W	GLTR	DCA
BRYBND	1000	**16**	28	34	-	75.71	59.81	**49.24**	-
CHAINWOOD	1000	**167**	-	712	-	2179.17	> 2500	**366.02**	-
COSINE	1000	18	19	**11**	-	62.59	26.90	**0.86**	-
CRAGGLVY	1000	16	16	16	25	12.17	4.54	**3.98**	47.69
DIXXMAANA	1500	10	10	10	10	7.92	2.85	**0.65**	31.26
DQRTIC	1000	-	77	**43**	63	-	514.51	756.49	**109.65**
FREUROTH	1000	11	11	11	-	10.29	3.47	**0.75**	-
GENROSE	1000	708	**703**	801	-	1350.43	649.72	**192.66**	-
MANCINO	100	16	16	16	-	12.84	9.52	**5.6**	-
NONCVXU2	1000	-	338	**327**	-	> 2500	605.91	**131.90**	-
NONCVXUN	1000	-	304	**276**	-	-	792.07	**471.39**	-
SENSORS	100	19	19	19	20	12.25	8.21	**2.44**	31.00
SINQUAD	5000	12	12	12	-	25.71	13.25	**1.88**	-
SPARSINE	1000	-	-	**29**	-	-	> 2500	**2082.05**	-

Table 2.9: Number of iterations and computation times (seconds) obtained by testing different TRS algorithms within Algorithm 2.8.1

and the computation time needed.

There are different reason for the absence of data in Table 2.9. First, we halted the computations if more than 2500 seconds were needed. Second, for the problems DQRTIC, NONCVXUN and SPARSINE and when using Algorithm 2.6.1 to solve the TRS (2.77), the computation was terminated because `eigs` was unable to return accurate eigenvalues. Third, when the DCA Algorithm was used to solve the TRS

(2.77), we terminated the computation if an optimal objective value of (2.77) was greater than zero, indicating a lack of convergence of the DCA Algorithm.

On the problems we tested, the general tendency is that when the GLTR Algorithm is used within Algorithm 2.8.1, less iterations and less computation times are involved (see [10] for numerical examples where the GLTR Algorithm may take more iterations than the Rendl-Wolkowicz Algorithm). Furthermore, only while using the GLTR Algorithm were we able to solve all of the problems within the 2500 seconds time limit.

If we compare the results obtained using Algorithm 2.6.1 and the Rendl-Wolkowicz Algorithm, we observe that Algorithm 2.8.1, with the Rendl-Wolkowicz Algorithm used to solve the TRS, leads, as it should be, to a faster and more robust method (since `eigs` never failed in the Rendl-Wolkowicz algorithm to compute the eigenvalues accurately). However, the results we obtained with Algorithm 2.6.1 were usually of the same order as the one obtained with the Rendl-Wolkowicz Algorithm.

Not surprisingly, Algorithm 2.8.1 combined with the DCA Algorithm did not prove to be robust nor fast, except only for DQRTIC. In our experience, this is caused by the slow convergence of the latter algorithm. Often, after 5000 iterations of the DCA Algorithm (which was the bound given) on the TRS (2.77), the approximate solution obtained did not have an objective value less than zero, the trivial upper bound on the optimal objective value.

2.10.3 Algorithm 2.9.1 for Constrained Optimization

In this section, we give results obtained by applying Algorithm 2.9.1 to solved constrained problems of the type (2.78). Our test problems are taken from the CUTEr [18] problem set. We have considered the following algorithms to find the approximate solutions of the TRS (2.86): the Moré-Sorensen Algorithm, Algorithm 2.6.1 and the GLTR Algorithm (the DCA Algorithm was initially considered as well, but

failed within Algorithm 2.9.1 to give a convergent method on any of the problems we tested). In Algorithm 2.6.1 and the Rendl-Wolkowicz Algorithm, we stopped when the relative duality gap was less than

$$\epsilon_1 = \max(10^{-12}, \min(10^{-6}, \mu_k^{1.5})).$$

The Moré-Sorensen Algorithm was halted when

$$q(\tilde{x}) < (1-\sigma)^2 q^*,$$

with $\sigma = \epsilon_1/\sqrt{1+\epsilon_1}$. This choice is made so that we hope solutions in both algorithms have the same relative accuracy. Again, as in Section §2.10.2, when using the GLTR Algorithm, we consider the couple $(s_{k,j}, \lambda_{k,j})$ of approximate solution and corresponding Lagrange multiplier a reasonable approximation of the optimal solution of (2.86) if

$$\|(N^T W_{k,j} N - \lambda_{k,j} I)s_{k,j} + N^T(\nabla f(x_{k,j}) - \mu_k J_{k,j}^T C_{k,j}^{-1} e)\| < \sqrt{\epsilon_1}$$

or if 10 iterations were done beyond the Steihaug-Toint point. Finally, the stopping tolerance *tol* in Algorithm 2.9.1 was set to 10^{-2}.

The problems we have chosen to use in our tests are of small and medium size. The main reason for this is that the matrix $N^T W_{k,j} N$ which appears in (2.86) will be dense. We have thus chosen to compute the required eigenvalues in Algorithm 2.6.1 with the MATLAB function `eig`.

All problems considered have linear constraints and most have a quadratic objective. The exceptions are HIMMELBI and SSEBLIN (this last problem has a linear objective). Bound constraints on the variables were treated just as the other linear constraints (although [7] suggest it is possible to handle these constraints in a special way). We chose as an initial x-iterate, x_0, the one suggested by CUTEr. However, we had to choose a different initial solution x_0 if the constraint $Ax_0 = b$ was not satisfied.

The new x_0 was obtained by solving a linear program with zero objective. We also altered slightly the bounds c_l and c_u on the constraints if $c_l < c(x_0) < c_u$ was not satisfied.

In Table 2.10.3, we report the computation time in seconds and the total number of inner iterations (inner it) needed to converge (because of our choice of $\mu_0 = 10$ and tol $= 10^{-2}$, convergence occurred after 4 outer iterations). For each problem, V indicates the number of variables, LE, the number of linear equality constraints and LI, the number of inequality constraints.

Problem	V	LE	LI	Inner iterations				Computation time			
				M-S	Alg. 2.6.1	R-W	GLTR	M-S	Alg. 2.6.1	R-W	GLTR
DUALC1	9	1	232	188	**143**	**143**	159	**7.22**	10.49	12.35	8.10
DUALC2	7	1	242	121	102	102	101	**4.30**	7.95	8.07	4.47
DUALC5	8	1	293	29	20	20	20	1.27	1.19	1.48	**1.10**
HATFLDH	4	0	21	38	36	**29**	**29**	0.89	2.13	1.50	**0.72**
HIMMELBI	100	0	212	**62**	132	91	100	**5.95**	43.21	17.08	9.49
PRIMALC2	231	0	236	**81**	113	111	> 1000	**23.64**	218.78	199.60	-
PRIMALC5	287	0	286	**65**	115	88	> 1000	**29.12**	844.44	286.20	-
QPCBOEI1	384	9	971	60	104	61	63	**84.61**	1234.40	286.40	289.76
QPCBOEI2	143	4	378	81	162	**71**	**71**	**14.93**	112.14	22.02	353.40
QPCSTAIR	467	209	696	40	48	37	37	**24.67**	148.97	67.14	63.02
QPNBOEI2	143	4	378	458	658	**384**	> 1000	**94.75**	636.74	123.05	-
QPNSTAIR	467	209	696	97	104	96	98	64.47	740.48	200.60	**32.00**

Table 2.10: Number of inner iterations and computation times obtained by testing different TRS algorithms within Algorithm 2.9.1

Using the Moré-Sorensen Algorithm to solve the TRS (2.86) within Algorithm

2.9.1 gives the best results in terms of computation time. This is in some sense not surprising since the TRS solved are dense and `eig` was used in the computation of eigenvalues. What is interesting however, is that GLTR failed to converge after 1000 inner iterations on three problems, where Algorithm 2.6.1 and the Rendl-Wolkowicz Algorithm succeeded (on the other hand the results in [7] are much better than what we were able to obtain with our personal MATLAB implementation of Algorithm 2.9.1 and the GLTR Algorithm). However, when Algorithm 2.9.1 does converge when the GLTR Algorithm is used to solve the TRS, the computation times are usually smaller than those obtained when Algorithm 2.6.1 or the Rendl-Wolkowicz Algorithm are used. There thus appears to be a trade-off in this case between robustness of the method and speed.

Chapter 3

Local-Nonglobal Minimizer

3.1 Background Results

We start by surveying the work of Martínez [29]. Hence, the reader is referred to this paper for the corresponding proofs of the lemmas and theorems which appear in this section.

The first theorem states the classical necessary optimality conditions for local minimizers of (2.1) and (2.2).

Theorem 3.1. *1. Assume that x^* is a local minimizer of (2.2). Then there exists a unique $\lambda^* \in \mathbb{R}$ such that*

$$(A - \lambda^* I)x^* = a \tag{3.1}$$

and

$$w^T(A - \lambda^* I)w \geq 0 \tag{3.2}$$

for all $w \in \mathbb{R}^n$ such that $w^T x^* = 0$.

2. Assume that x^* is a local minimizer of (2.1). If $\|x^*\| = 1$, there exists $\lambda^* \leq 0$ such that (3.1) and (3.2) hold. If $\|x^*\| < 1$, then x^* is a global minimizer, equation (3.1) holds with $\lambda^* = 0$ and A is positive semidefinite.

3. *Assume that* $\|x^*\| = 1$, $\lambda^* \in \mathbb{R}$ *satisfies* (3.1), *and*

$$w^T(A - \lambda^* I)w > 0 \tag{3.3}$$

for all $w \neq 0$ *such that* $w^T x^* = 0$. *Then* x^* *is a strict local minimizer of* (2.2).

4. *Assume that* $\|x^*\| = 1$ *and that* $\lambda^* < 0$ *satisfies* (3.1) *and* (3.3). *Then* x^* *is a strict local minimizer of* (2.1). □

Recall from the optimality condition (2.5b) that the Lagrange multiplier for a global minimizer of problem (2.1) lies in the interval $(-\infty, \lambda_1(A)]$. There exists bounds on the Lagrange multiplier of a local-nonglobal minimizer which depends as well on eigenvalues of A.

Lemma 3.1. *If* x^* *is a local-nonglobal minimizer of* (2.1) *or* (2.2), *then* (3.1) *holds with* $\lambda^* \in (\lambda_1(A), \lambda_2(A))$. □

Global minimizers of problems (2.1) or (2.2) always exist, since a continuous function is minimized over a compact set. However, it is not always the case that a local-nonglobal minimizer exists for one of these two problems. In particular, we have the two following cases for which no such point exists. The first case is an obvious consequence of the previous lemma.

Corollary 3.1. *If* $\lambda_1(A) = \lambda_2(A)$ *then there are no local-nonglobal minimizer of problems* (2.1) *or* (2.2). □

Lemma 3.2. *If* a *is orthogonal to an eigenvector of* A *for the eigenvalue* $\lambda_1(A)$, *then there are no local-nonglobal minimizer of problems* (2.1) *or* (2.2). □

When a is orthogonal to all eigenvectors of A for the eigenvalue $\lambda_1(A)$, then the hard case occurs. However, the previous lemma states this cannot happen if a local-nonglobal minimizer exists.

Define

$$\varphi(\lambda) := \|(A - \lambda I)^{-1}a\|^2. \tag{3.4}$$

Let $A = QDQ^T$ be an orthonormal diagonalization of A, i.e. the columns of Q are orthonormal eigenvectors of A and D is a diagonal matrix with the eigenvalues of A on its diagonal ordered increasingly such that $D_{11} = \lambda_1(A)$. Also let $\bar{a} := Q^T a$. The function φ and its derivatives are given by the following formulas:

$$\varphi(\lambda) = \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i(A) - \lambda)^2},$$

$$\varphi'(\lambda) = 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i(A) - \lambda)^3}, \tag{3.5}$$

$$\varphi''(\lambda) = 6\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i(A) - \lambda)^4}. \tag{3.6}$$

Suppose $\lambda_1(A) < \lambda_2(A)$ and $\bar{a}_1 \neq 0$. Note from (3.6) that φ is a strictly convex function over the interval $(\lambda_1(A), \lambda_2(A))$. Therefore the equation $\varphi(\lambda) = 1$ has at most two roots in $(\lambda_1(A), \lambda_2(A))$ and the following theorem shows, that in the case two roots exists, only the smallest root may be a local-nonglobal minimizer of problems (2.1) or (2.2).

Theorem 3.2. *1. If x^* is a local-nonglobal minimizer of* (2.1) *or* (2.2)*, then* (3.1) *holds with $\lambda^* \in (\lambda_1(A), \lambda_2(A))$ and $\varphi'(\lambda^*) \leq 0$. If x^* is a local-nonglobal minimizer of* (2.1) *then $\lambda^* \leq 0$.*

2. There exists at most one local-nonglobal minimizer of (2.1) *or* (2.2)*.*

3. If $\|x^*\| = 1$, (3.1) *holds for some $\lambda^* \in (\lambda_1(A), \lambda_2(A))$ and $\varphi'(\lambda^*) < 0$, then x^* is a strict local-nonglobal minimizer of* (2.2)*. If in addition, $\lambda^* < 0$, x^* is also a strict local-nonglobal minimizer of* (2.1)*.* □

3.2 Computing a Local-Nonglobal Minimizer

As mentioned in the introduction, our intention is to build on the theory behind the algorithms of Chapter 2 and the Rendl-Wolkowicz Algorithm [39], which compute a global minimizer of problem (2.2), in order to compute a local-nonglobal minimizer of problems (2.1) and (2.2). In these algorithms, one needs at each iteration the smallest eigenvalue of a parameterized matrix. In the algorithms of this chapter, the first two eigenvalues are relevant. The functions of interest for our algorithms will be similar to the ones used in Algorithms 2.6.1 and 2.6.2, except that the first eigenvalue of the parameterized matrices may need to be replaced by the second one.

We assume throughout this section the following assumptions hold:

Assumption 3.2.1. $\lambda_1(A) < \lambda_2(A)$;

Assumption 3.2.2. $\bar{a}_1 \neq 0$.

These are justified by Corollary 3.1 and Lemma 3.2. Note also that whenever we refer to the matrix B in this chapter, we mean $B = A - \bar{\lambda}I$ with $\bar{\lambda} = \lambda_1(A) - \|a\|$. This choice of $\bar{\lambda}$ and Assumption 3.2.1 guarantee that the equations (2.7) will hold.

3.2.1 Computing a Local-Nonglobal Minimizer: First Method

In Section §2.2, we showed solving problem (2.2) is equivalent to finding the largest volume ellipsoid E_{r_G} among the ellipsoids E_r contained in the unit ball. Equivalence is in the sense that E_{r_G} intersects the unit sphere at a global minimizer x_G for (2.2). Similarly, a local-nonglobal minimizer x_L of problem (2.2) lies in the intersection of an ellipsoid E_{r_L} and the unit sphere, for some $r_L > r_G$. This ellipsoid is not contained in the unit ball, but points which are element of E_{r_L} and close enough to x_L are contained in the unit ball. This is illustrated in Figure 3.1. Recall from problem (2.13) that in order to compute a global minimizer we maximize the function $f(\gamma) = \gamma\lambda_1(B(\gamma))$.

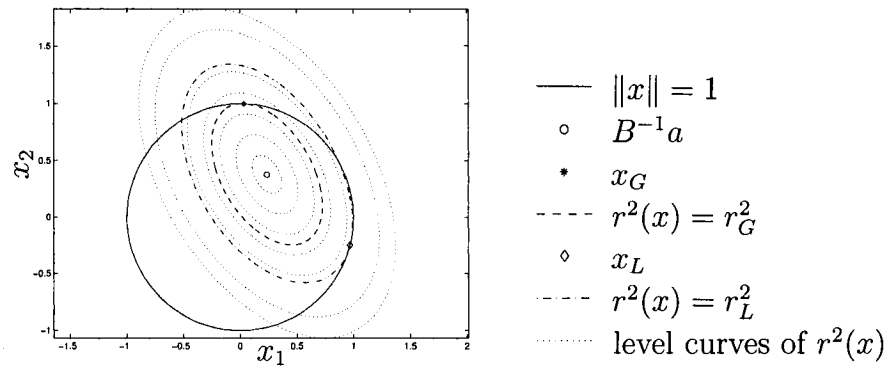

Figure 3.1: This figure illustrates a local-nonglobal minimizer x_L for problem (2.2) intersects the unit sphere and the ellipsoid E_{r_L} which is locally contained in the unit ball at x_L. Here the global minimizer of problem (2.2) is x_G.

We shall see that for finding a local-nonglobal minimizer of problems (2.1) or (2.2), a new function g related to the function f is involved. Corollary 2.1 and Lemma 3.1 motivate the definition of this new function.

For problem (2.2), the Lagrange multiplier of a global optimal solution lies in the interval $(-\infty, \lambda_1(A))$. It was shown in Lemmas 2.3 and 2.4 that the function $\lambda_1(B(\gamma)) + \bar{\lambda}$ of a single variable γ maps the interval $(-\infty, 1)$ onto the interval $(-\infty, \lambda_1(A))$. Furthermore, $\lambda(B(\hat{\gamma})) + \bar{\lambda}$ is the Lagrange multiplier, where $\hat{\gamma}$ is the optimal solution for problem (2.13). On the other hand, recall from Lemma 3.1 that the Lagrange multiplier of a local-nonglobal minimizer of problem (2.2) lies in the interval $(\lambda_1(A), \lambda_2(A))$. Notice from the definition of B that $\lambda_i(B) = \lambda_i(A) - \bar{\lambda}$ for $i = 1 \ldots n$ and, from Assumption 3.2.1, $\lambda_1(B) < \lambda_2(B)$. Thus, Corollary 2.1 suggests investigating the function defined as $\lambda_1(B(\gamma)) + \bar{\lambda} \in (\lambda_1(A), \lambda_2(A))$ for $\gamma > 1$ and $\lambda_2(B(\gamma)) + \bar{\lambda} \in (\lambda_1(A), \lambda_2(A))$ for $\gamma < 1$. This is done in the next lemma.

Let

$$\begin{aligned} \mathcal{K} &:= \{k : \lambda_k(B) = \lambda_2(B)\}, && (3.7a) \\ \mathcal{J} &:= \{j : \bar{a}_j \neq 0\}, && (3.7b) \\ d(\lambda) &:= \sum_{j\in\mathcal{J}} \frac{\bar{a}_j^2}{\lambda_j(B)(\lambda_j(B)-\lambda)}. \end{aligned}$$

Note Assumptions 3.2.1 and 3.2.2 may be stated as $1 \in \mathcal{K}^c \cap \mathcal{J}$, where $\mathcal{K}^c$ is the complement of the set $\mathcal{K}$.

Lemma 3.3. *Let*

$$\lambda(\gamma) := \begin{cases} \lambda_2(B(\gamma)) & \text{for } \gamma < 1, \\ \hat{\lambda} & \text{for } \gamma = 1, \\ \lambda_1(B(\gamma)) & \text{for } \gamma > 1, \end{cases}$$

where $\hat{\lambda} = \lambda_2(B)$ if $\mathcal{K} \cap \mathcal{J} = \varnothing$ and $d(\lambda_2(B)) \leq 0$; otherwise, $\hat{\lambda}$ is the unique value in the interval $(\lambda_1(B), \lambda_2(B))$ to satisfy $d(\hat{\lambda}) = 0$.

1. *If $\mathcal{K} \cap \mathcal{J} \neq \varnothing$, $\lambda(\gamma)$ is infinitely differentiable and satisfies $d(\lambda(\gamma)) = 1 - \gamma$. Moreover,*

$$\lambda'(\gamma) = \frac{-1}{d'(\lambda(\gamma))} \quad \text{and} \quad \lambda''(\gamma) = \frac{-d''(\lambda(\gamma))}{[d'(\lambda(\gamma))]^3} \tag{3.8}$$

   for all $\gamma \in \mathbb{R}$.

2. *If $\mathcal{K} \cap \mathcal{J} = \varnothing$, $\lambda(\gamma)$ is continuous and infinitely differentiable for $\gamma \in \mathbb{R} \setminus \{1 - d(\lambda_2(B))\}$.*

   (a) *For $\gamma > 1 - d(\lambda_2(B))$, $\lambda(\gamma)$ satisfies $d(\lambda(\gamma)) = 1 - \gamma$ and*

$$\lambda'(\gamma) = \frac{-1}{d'(\lambda(\gamma))} \quad \text{and} \quad \lambda''(\gamma) = \frac{-d''(\lambda(\gamma))}{[d'(\lambda(\gamma))]^3}. \tag{3.9}$$

   (b) *For $\gamma < 1 - d(\lambda_2(B))$, $\lambda(\gamma) = \lambda_2(B)$, $\lambda'(\gamma) = 0$ and $\lambda''(\gamma) = 0$.*

(c) for $\gamma = 1 - d(\lambda_2(B))$, $\lambda(\gamma) = \lambda_2(B)$ *and satisfies* $d(\lambda(\gamma)) = 1 - \gamma$; *the right handed and left handed derivatives are given respectively by*

$$\lambda'(\gamma^+) = \frac{-1}{d'(\lambda_2(B))}, \quad \lambda''(\gamma^+) = \frac{-d''(\lambda_2(B))}{[d'(\lambda_2(B))]^3}, \tag{3.10a}$$

$$\lambda'(\gamma^-) = 0, \qquad \lambda''(\gamma^-) = 0. \tag{3.10b}$$

Proof. 1. For $\gamma \neq 1$, we have

$$\begin{aligned}
\det(B(\gamma) - \lambda I) &= \det\left((B - \lambda I)\left(I - \frac{1}{1-\gamma}(B - \lambda I)^{-1} B^{-1/2} a a^T B^{-1/2}\right)\right), \\
&= \det(B - \lambda I)\left(1 - \frac{1}{1-\gamma} a^T B^{-1/2}(B - \lambda I)^{-1} B^{-1/2} a\right), \\
&= \prod_{j=1}^{n}(\lambda_j(B) - \lambda)\left(1 - \frac{1}{1-\gamma}\sum_{j \in \mathcal{J}} \frac{\bar{a}_j^2}{\lambda_j(B)(\lambda_j(B) - \lambda)}\right) \quad (3.11a) \\
&= \prod_{j=1}^{n}(\lambda_j(B) - \lambda)\left(1 - \frac{1}{1-\gamma} d(\lambda)\right), \quad (3.11b)
\end{aligned}$$

where the second equality follows from Golub and Van Loan [15]. Since $\mathcal{K} \cap \mathcal{J} \neq \varnothing$ and $\bar{a}_1 \neq 0$, then

$$\lim_{\lambda \searrow \lambda_1(B)} d(\lambda) = -\infty \quad \text{and} \quad \lim_{\lambda \nearrow \lambda_2(B)} d(\lambda) = \infty.$$

Furthermore,

$$d'(\lambda) = \sum_{j \in \mathcal{J}} \frac{\bar{a}_j^2}{\lambda_j(B)(\lambda_j(B) - \lambda)^2} > 0.$$

Therefore, for all $\gamma \in \mathbb{R}$, $d^{-1}(1 - \gamma)$ is well defined, where $d^{-1}(1 - \gamma) \in (\lambda_1(B), \lambda_2(B))$. Moreover, (3.11b) shows it is an eigenvalue of $B(\gamma)$. From Corollary 2.1, this shows

$$d^{-1}(1 - \gamma) = \begin{cases} \lambda_2(B(\gamma)) & \text{for } \gamma < 1, \\ \hat{\lambda} & \text{for } \gamma = 1, \\ \lambda_1(B(\gamma)) & \text{for } \gamma > 1. \end{cases}$$

Hence, $\lambda(\gamma) = d^{-1}(1 - \gamma)$ and is infinitely differentiable. Equations (3.8) are obtained by implicit differentiation.

2. Since $\mathcal{K} \cap \mathcal{J} = \varnothing$, then $d(\lambda_2(B))$ is well defined. Let $\gamma \neq 1$. By equation (3.11b), $\lambda_2(B)$ is an eigenvalue of $B(\gamma)$. Also, Assumptions 3.2.1 and 3.2.2 imply $\lambda_1(B)$ is not an eigenvalue of $B(\gamma)$, since from equation (3.11a) we obtain

$$\lim_{\lambda \to \lambda_1(B)} \det(B(\gamma) - \lambda I) = -\prod_{j=2}^{n} (\lambda_j(B) - \lambda_1(B)) \left(\frac{\bar{a}_1^2}{(1-\gamma)\lambda_1(B)} \right) \neq 0.$$

Note again $d(\lambda)$ is strictly increasing for $\lambda \in (\lambda_1(B), \lambda_2(B)]$ and therefore

$$d(\lambda) < d(\lambda_2(B)) \text{ for } \lambda \in (\lambda_1(B), \lambda_2(B)). \tag{3.12}$$

(a) If $1 - \gamma < d(\lambda_2(B))$, then $d^{-1}(1-\gamma)$ is well defined, where $d^{-1}(1-\gamma) \in (\lambda_1(B), \lambda_2(B))$. The rest of the proof is similar to the proof of item 1.

(b) and (c) If $1-\gamma \geq d(\lambda_2(B))$, by (3.12), $d(\lambda) < 1-\gamma$ for $\lambda \in (\lambda_1(B), \lambda_2(B))$. Therefore, there are no eigenvalues of $B(\gamma)$ in the interval $[\lambda_1(B), \lambda_2(B))$ and, from Corollary 2.1,

$$\lambda_2(B) = \begin{cases} \lambda_2(B(\gamma)) & \text{for } \gamma < 1, \\ \hat{\lambda} & \text{for } \gamma = 1, \\ \lambda_1(B(\gamma)) & \text{for } \gamma > 1. \end{cases}$$

Thus $\lambda(\gamma) = \lambda_2(B)$. In particular, the derivatives of $\lambda(\gamma)$ for $\gamma < 1 - d(\lambda_2(B))$ are zero and equations (3.10b) hold. Note finally $1 - \gamma = d(\lambda(\gamma))$ for $\gamma \geq 1 - d(\lambda_2(B))$ and thus (3.10a) holds.

□

Corollary 3.2. *For $\gamma \in \mathbb{R}$, $\lambda(\gamma) > \lambda_1(B)$ and $\lim_{\gamma \to \infty} \lambda(\gamma) = \lambda_1(B)$. Moreover,*

1. *if $\mathcal{K} \cap \mathcal{J} \neq \varnothing$, $\lambda(\gamma) < \lambda_2(B)$ and $\lim_{\gamma \to -\infty} \lambda(\gamma) = \lambda_2(B)$.*

2. *If $\mathcal{K} \cap \mathcal{J} = \varnothing$,*

   (a) $\lambda(\gamma) = \lambda_2(B)$ for $\gamma \leq 1 - d(\lambda_2(B))$,

(b) $\lambda(\gamma) < \lambda_2(B)$ *for* $\gamma > 1 - d(\lambda_2(B))$. □

Recall in problem (2.13) the function $f(\gamma) = \gamma\lambda_1(B(\gamma))$ is maximized over the interval $(-\infty, 1)$ and the optimal γ^* is used to construct a global minimizer for problem (2.2). This motivates the definition of the following function: $g(\gamma) := \gamma\lambda(\gamma)$. Its domain is $\mathbb{R}$. An obvious question is if an optimum of the function g is related to a local-nonglobal minimizer of problem (2.1) or (2.2). This will be answered in Theorems 3.4 and 3.5. As a first step toward an answer, the next lemma is concerned with the first derivative of g.

Let

$$\Gamma := \begin{cases} \mathbb{R}, & \text{if } \mathcal{K} \cap \mathcal{J} \neq \varnothing, \\ (1 - d(\lambda_2(B), \infty), & \text{if } \mathcal{K} \cap \mathcal{J} = \varnothing. \end{cases}$$

Note, from Lemma 3.3 and Corollary 3.2,

$$\Gamma = \{\gamma : \lambda(\gamma) \in (\lambda_1(B), \lambda_2(B))\}, \tag{3.13}$$

$$\lambda'(\gamma) < 0 \quad \text{for} \quad \gamma \in \Gamma. \tag{3.14}$$

Lemma 3.4. *For* $\gamma \in \Gamma \setminus \{1\}$, *let the vector* $v \in \mathbb{R}^n$ *satisfy the equation*

$$\left(B^2 - \frac{1}{1-\gamma} aa^T\right) v = \lambda(\gamma) B v. \tag{3.15}$$

Note v *depends on* γ. *Then we may write the derivative of* g *as*

$$g'(\gamma) = \lambda(\gamma) - \frac{\gamma}{v^T B v} \left(\frac{a^T v}{1-\gamma}\right)^2. \tag{3.16}$$

Proof. Using Corollary 2.1 and equation (3.13), we obtain

$$\begin{aligned} \lambda_1(B) < \lambda_1(B(\gamma)) < \lambda_2(B) \leq \lambda_2(B(\gamma)) \quad &\text{if} \quad \gamma > 1, \\ \lambda_1(B(\gamma)) \leq \lambda_1(B) < \lambda_2(B(\gamma)) < \lambda_2(B) \leq \lambda_3(B(\gamma)) \quad &\text{if} \quad \gamma < 1. \end{aligned}$$

Hence $\lambda(\gamma)$ is an eigenvalue of $B(\gamma)$ of multiplicity one and it is easy to see $\frac{B^{1/2}v}{\|B^{1/2}v\|}$ is a corresponding unit norm eigenvector. Therefore (see e.g. Horn and Johnson [23,

pp.185]),

$$\begin{aligned} g'(\gamma) &= \lambda(\gamma) - \gamma \left(\frac{B^{1/2}v}{\|B^{1/2}v\|}\right)^T \left(\frac{1}{(1-\gamma)^2} B^{-1/2} a a^T B^{-1/2}\right) \left(\frac{B^{1/2}v}{\|B^{1/2}v\|}\right), \\ &= \lambda(\gamma) - \frac{\gamma}{v^T B v} \left(\frac{a^T v}{1-\gamma}\right)^2. \end{aligned}$$

□

When $\mathcal{K} \cap \mathcal{J} = \varnothing$ and $\gamma < 1 - d(\lambda_2(B))$, then $g'(\gamma) = \lambda_2(B)$. For $\gamma = 1 - d(\lambda_2(B))$, g may not be differentiable since it is possible to show the multiplicity of $\lambda(\gamma)$ is changing. As seen in Chapter 2, a similar phenomena occurs in the hard case and needs to be well understood when one is interested in computing a global minimum. However, when computing a local-nonglobal minimizer, this phenomena has little consequence on our algorithm and a detailed analysis similar to the one done in Lemma 2.5 is not necessary.

For $\gamma \in \Gamma$, define

$$x(\gamma) := \begin{cases} \frac{1-\gamma}{a^T v} B v & \text{if } \gamma \neq 1, \\ (A - (\lambda(1) + \bar{\lambda})I)^{-1} a & \text{if } \gamma = 1, \end{cases} \tag{3.17}$$

where v satisfies (3.15). Note $a^T v \neq 0$, otherwise this would imply $\lambda(\gamma)$ is an eigenvalue of B. Rewriting equation (3.15) and substituting B for $A - \bar{\lambda} I$, we see $x(\gamma)$ satisfies the first order necessary condition (3.1), i.e.

$$(A - (\lambda(\gamma) + \bar{\lambda})I) x(\gamma) = a. \tag{3.18}$$

For $\gamma \in \Gamma$, $\lambda(\gamma) + \bar{\lambda} \in (\lambda_1(A), \lambda_2(A))$. Therefore $A - (\lambda(\gamma) + \bar{\lambda})I$ is invertible, and we may write $\|x(\gamma)\|^2$ as

$$\|x(\gamma)\|^2 = \varphi(\lambda(\gamma) + \bar{\lambda}). \tag{3.19}$$

It follows

$$\frac{d\|x(\gamma)\|^2}{d\gamma} = \frac{d\varphi(\lambda(\gamma) + \bar{\lambda})}{d\lambda} \frac{d\lambda(\gamma)}{\gamma}. \tag{3.20}$$

The Martínez Algorithm [29] which is used to compute a local-nonglobal minimizer of problems (2.1) and (2.2) finds a root of the function $\varphi(\lambda) - 1$ in the interval $(\lambda_1(A), \lambda_2(A))$. As we shall see in Section §3.2.1, our algorithm finds a root of the function $\|x(\gamma)\|^2 - 1$ in the interval Γ. We may immediately derive the equivalent of Theorem 3.2.

Theorem 3.3. *1. If x^* is a local-nonglobal minimizer of problems* (2.1) *or* (2.2), *then* (3.1) *holds with $\lambda^* \in (\lambda_1(A), \lambda_2(A))$. Let γ^* be the unique solution to $\lambda(\gamma) + \bar{\lambda} = \lambda^*$, then $x^* = x(\gamma^*)$ and $\frac{d\|x(\gamma^*)\|^2}{d\gamma} \geq 0$. If x^* is a local-nonglobal minimizer of* (2.1) *then $\lambda(\gamma^*) + \bar{\lambda} \leq 0$.*

2. If, for $\gamma^* \in \Gamma$, $\|x(\gamma^*)\| = 1$ and $\frac{d\|x(\gamma^*)\|^2}{d\gamma} > 0$, then $x(\gamma^*)$ is a strict local-nonglobal minimizer of (2.2). *If in addition, $\lambda(\gamma^*) + \bar{\lambda} < 0$, $x(\gamma^*)$ is also a strict local-nonglobal minimizer of* (2.1).

3. For $\gamma \in \Gamma \cap \{\gamma : \frac{d\|x(\gamma)\|^2}{d\gamma} > 0\}$, $x(\gamma)$ is a strict local non-global minimizer of

$$\begin{aligned} \min_x \quad & x^T A x - 2a^T x \\ s.t. \quad & \|x\| = \|x(\gamma)\| \end{aligned} \tag{3.21}$$

with Lagrange multiplier $\lambda(\gamma) + \bar{\lambda}$. In addition, if $\lambda(\gamma) + \bar{\lambda} < 0$, then $x(\gamma)$ is a strict local-nonglobal minimizer of

$$\begin{aligned} \min_x \quad & x^T A x - 2a^T x \\ s.t. \quad & \|x\| \leq \|x(\gamma)\|. \end{aligned} \tag{3.22}$$

Proof. Since $\gamma^* \in \Gamma$, the proofs of items 1 and 2 follow from Theorem 3.2 and equations (3.14), (3.19) and (3.20). To prove item 3, fix $\gamma \in \Gamma$ and let $\delta := \|x(\gamma)\|$. Note $x(\gamma)$ is a local-nonglobal minimizer of problem (3.21) if and only if $x(\gamma; \delta) := x(\gamma)/\delta$ is a local-nonglobal minimizer of

$$\begin{aligned} \min_x \quad & x^T(\delta^2 A)x - 2(\delta a)^T x \\ \text{s.t.} \quad & \|x\| = 1. \end{aligned} \tag{3.23}$$

Now we may write $\varphi(\lambda;\delta) := \|((\delta^2 A) - \lambda I)^{-1}(\delta a)\|^2$ as

$$\varphi(\lambda;\delta) = \sum_{i=1}^{n} \frac{\delta^2 \bar{a}_i^2}{(\delta^2\lambda_i(A) - \lambda)^2}, \tag{3.24}$$

and its derivative as

$$\varphi'(\lambda;\delta) = 2\sum_{i=1}^{n} \frac{\delta^2 \bar{a}_i^2}{(\delta^2\lambda_i(A) - \lambda)^3}.$$

For $\gamma \in \Gamma \cap \{\gamma : \frac{d\|x(\gamma)\|^2}{d\gamma} > 0\}$, it follows by equations (3.14) and (3.20) that

$$\frac{d\varphi(\lambda(\gamma) + \bar{\lambda})}{d\lambda} < 0. \tag{3.25}$$

By item 3 of Theorem 3.2, $x(\gamma;\delta)$ is a local-nonglobal minimizer of problem (3.23), since

$$((\delta^2 A) - (\delta^2(\lambda(\gamma) + \bar{\lambda}))I)x(\gamma;\delta) = \delta a,$$

since $\lambda_1(\delta^2 A) < \delta^2(\lambda(\gamma) + \bar{\lambda}) < \lambda_2(\delta^2 A)$ and since, from inequality (3.25),

$$\varphi'(\delta^2(\lambda(\gamma) + \bar{\lambda});\delta) = \varphi'(\lambda(\gamma) + \bar{\lambda}) < 0.$$

□

From equation (3.6) we easily see φ is a strictly convex function on the open interval $(\lambda_1(A), \lambda_2(A))$. The Martínez Algorithm takes advantage of this property. The following lemma shows $\|x(\gamma)\|^2$ is also strictly convex over Γ. Convexity will play a main role in the convergence analysis of our algorithm, since the secant method will be used to find a root of the function $\|x(\gamma)\|^2 - 1$.

Lemma 3.5. *Consider the function* $\|x(\gamma)\|^2$ *with domain* Γ. *Then it is an infinitely differentiable strictly convex function and* $\lim_{\gamma\to\infty} \|x(\gamma)\|^2 = \infty$.

Proof. Since $\varphi(\lambda)$ and $\lambda(\gamma)$ are infinitely differentiable respectively on the intervals $(\lambda_1(A), \lambda_2(A))$ and Γ, and $\lambda(\gamma) + \bar{\lambda} \in (\lambda_1(A), \lambda_2(A))$, then infinite differentiability follows from (3.19). By Corollary 3.2, $\lim_{\gamma\to\infty} \lambda(\gamma) + \bar{\lambda} = \lambda_1(A)$ and $\lambda(\gamma) + \bar{\lambda} >$

$\lambda_1(A)$, and, by Assumption 3.2.2, $\lim_{\lambda \searrow \lambda_1(A)} \varphi(\lambda) = \infty$. Thus, using equation (3.19), $\lim_{\gamma \to \infty} \|x(\gamma)\|^2 = \infty$.

All that is left to prove is strict convexity. For simplicity, let $\lambda_i = \lambda_i(B)$, for $i = 1, \ldots, n$ and let $\lambda_\gamma = \lambda(\gamma)$. There are two cases to consider.

1. *Case 1:* $\bar{a}_1 \neq 0$ *and* $\bar{a}_j = 0$ *for* $j = 2, \ldots, n$. We have in this case

$$\begin{aligned} \|x(\gamma)\|^2 &= \frac{\bar{a}_1^2}{(\lambda_1 - \lambda_\gamma)^2}, \\ \frac{d\|x(\gamma)\|^2}{d\gamma} &= \frac{2\bar{a}_1^2}{(\lambda_1 - \lambda_\gamma)^3}\lambda'(\gamma) = -\frac{2\lambda_1}{\lambda_1 - \lambda_\gamma}, \end{aligned}$$

where equations (3.8) and (3.9) were used to obtain the first derivative. Thus, using equation (3.14),

$$\frac{d^2\|x(\gamma)\|^2}{d\gamma^2} = -\frac{2\lambda_1}{(\lambda_1 - \lambda_\gamma)^2}\lambda'(\gamma) > 0.$$

2. *Case 2:* $\exists j \geq 2$ *such that* $\bar{a}_1\bar{a}_j \neq 0$. We have, using once again equations (3.8) and (3.9),

$$\begin{aligned} \|x(\gamma)\|^2 &= \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_\gamma)^2}, \\ \frac{d\|x(\gamma)\|^2}{d\gamma} &= 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_\gamma)^3}\lambda'(\gamma) = \frac{-2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_\gamma)^3}}{\sum_{i=1}^{n} \frac{\bar{a}_i^2}{\lambda_i(\lambda_i - \lambda_\gamma)^2}}, \\ \frac{d^2\|x(\gamma)\|^2}{d\gamma^2} &= \frac{\left(-6\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_\gamma)^4}\sum_{i=1}^{n} \frac{\bar{a}_i^2}{\lambda_i(\lambda_i - \lambda_\gamma)^2} + 4\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_\gamma)^3}\sum_{i=1}^{n} \frac{\bar{a}_i^2}{\lambda_i(\lambda_i - \lambda_\gamma)^3}\right)\lambda'(\gamma)}{\left(\sum_{i=1}^{n} \frac{\bar{a}_i^2}{\lambda_i(\lambda_i - \lambda_\gamma)^2}\right)^2}. \end{aligned}$$

From equations (3.13) and (3.14), our result is proved if we can show for all $\lambda \in (\lambda_1, \lambda_2)$, that

$$-3\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^4}\sum_{j=1}^{n} \frac{\bar{a}_j^2}{\lambda_j(\lambda_j - \lambda)^2} + 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^3}\sum_{j=1}^{n} \frac{\bar{a}_j^2}{\lambda_j(\lambda_j - \lambda)^3}$$

is strictly negative. In fact, we prove the stronger statement, for $\lambda \in (\lambda_1, \lambda_2)$, that

$$-\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^4} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{\lambda_j(\lambda_j - \lambda)^2} + \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^3} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{\lambda_j(\lambda_j - \lambda)^3} \tag{3.26}$$

is strictly negative. We may rewrite (3.26) as

$$\sum_{i,j=1}^{n} \frac{\bar{a}_i^2 \bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2} \left(-1 + \frac{\lambda_i - \lambda}{\lambda_j - \lambda}\right) = \sum_{i,j=1}^{n} \frac{\bar{a}_i^2 \bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2} \left(\frac{\lambda_i - \lambda_j}{\lambda_j - \lambda}\right)$$
$$= \sum_{i,j=1, i \neq j}^{n} \frac{\bar{a}_i^2 \bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2} \left(\frac{\lambda_i - \lambda_j}{\lambda_j - \lambda}\right).$$

The previous sum may be rewritten as

$$\sum_{j=2}^{n} \left\{ \frac{\bar{a}_1^2 \bar{a}_j^2}{\lambda_j(\lambda_1 - \lambda)^4(\lambda_j - \lambda)^2} \left(\frac{\lambda_1 - \lambda_j}{\lambda_j - \lambda}\right) + \frac{\bar{a}_j^2 \bar{a}_1^2}{\lambda_1(\lambda_j - \lambda)^4(\lambda_1 - \lambda)^2} \left(\frac{\lambda_j - \lambda_1}{\lambda_1 - \lambda}\right) \right\} +$$
$$\sum_{i=2}^{n} \sum_{j>i} \left\{ \frac{\bar{a}_i^2 \bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2} \left(\frac{\lambda_i - \lambda_j}{\lambda_j - \lambda}\right) + \frac{\bar{a}_j^2 \bar{a}_i^2}{\lambda_i(\lambda_j - \lambda)^4(\lambda_i - \lambda)^2} \left(\frac{\lambda_j - \lambda_i}{\lambda_i - \lambda}\right) \right\}.$$

Recall, from Assumption 3.2.2, $\lambda_1 < \lambda_2$. Thus, the first sum is strictly negative for $\lambda \in (\lambda_1, \lambda_2)$, where we use there exists $j \geq 2$ such that $\bar{a}_1 \bar{a}_j \neq 0$. We next claim, for $2 \leq i \leq n-1$ and $i < j \leq n$, that

$$\frac{\bar{a}_i^2 \bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2} \left(\frac{\lambda_i - \lambda_j}{\lambda_j - \lambda}\right) + \frac{\bar{a}_j^2 \bar{a}_i^2}{\lambda_i(\lambda_j - \lambda)^4(\lambda_i - \lambda)^2} \left(\frac{\lambda_j - \lambda_i}{\lambda_i - \lambda}\right) \tag{3.27}$$

is negative. Indeed, if $\bar{a}_i \bar{a}_j = 0$ or $\lambda_i = \lambda_j$, it is trivial. Otherwise, $\bar{a}_i \bar{a}_j \neq 0$ and $\lambda_i < \lambda_j$ and (3.27) is negative if and only if

$$\frac{-1}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^3} + \frac{1}{\lambda_i(\lambda_j - \lambda)^4(\lambda_i - \lambda)^3}$$

is negative. Rewriting the last expression, we obtain

$$\frac{-\lambda(\lambda_j - \lambda_i)}{\lambda_i \lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^4},$$

which is negative, since $\lambda \geq \lambda_1 > 0$ and $\lambda_j > \lambda_i$. Thus (3.26) is strictly negative and $\|x(\gamma)\|^2$ is a strictly convex function for $\gamma \in \Gamma$. □

Similarly to Lemma 2.7, our next lemma shows $\|x(\gamma)\|$ is related to the first derivative of g.

Lemma 3.6. *Let $\gamma \in \Gamma$. Then*

$$\|x(\gamma)\| > \ (=, <)\ 1 \iff g'(\gamma) > \ (=, <)\ 0. \tag{3.28}$$

Proof. For $\gamma \in \Gamma \setminus \{1\}$, we have

$$\begin{aligned}
\|x(\gamma)\|^2 &= \left(\frac{1-\gamma}{a^T v}\right)^2 v^T B^2 v, \\
&= 1 - \gamma + \lambda(\gamma)\left(\frac{1-\gamma}{a^T v}\right)^2 v^T B v, && (3.29a) \\
&= 1 + g'(\gamma)\left(\frac{1-\gamma}{a^T v}\right)^2 v^T B v, && (3.29b) \\
&= 1 + g'(\gamma)\|x(\gamma)\|^2 \frac{v^T B v}{v^T B^2 v}, && (3.29c)
\end{aligned}$$

where (3.29a) follows from (3.15) and (3.29b), from (3.16). The conclusion follows by writing (3.29c) as

$$\|x(\gamma)\|^2 = \frac{1}{1 - g'(\gamma)\frac{v^T B v}{v^T B^2 v}} \tag{3.30}$$

and noting that in the case where $\gamma = 1 \in \Gamma$, the relation (3.28) holds as well by continuity of the functions g' and $\|x(\gamma)\|^2$. □

We are now ready with the next two theorems to relate a candidate for a local minimizer of the function g to a local-nonglobal minimizer of problem (2.1) or (2.2). The first one states that if a local-nonglobal minimizer of problem (2.1) or (2.2) exists, then there exists γ^* where the first and second order optimality conditions for a local minimizer of g are satisfied. The second one is almost its converse: if the first and second order sufficient optimality conditions for a local minimizer of g are satisfied at some γ^*, then $x(\gamma^*)$ is the local-nonglobal minimizer of problem (2.2). It is also a local-nonglobal minimizer of problem (2.1), if the sign of the Lagrange multiplier is strictly negative.

Theorem 3.4. *Suppose x^* is a local-nonglobal minimizer of* (2.2) *or* (2.1) *with corresponding Lagrange multiplier λ^*. Let γ^* be the solution to $\lambda(\gamma) + \bar{\lambda} = \lambda^*$. Then $g'(\gamma^*) = 0$ and $g''(\gamma^*) \geq 0$.*

Proof. By Lemma 3.1, $\lambda(\gamma^*) \in (\lambda_1(B), \lambda_2(B))$ and thus $\gamma^* \in \Gamma$. Theorem 3.3 gives $x^* = x(\gamma^*)$. The fact that $g'(\gamma^*) = 0$ follows from the feasibility of x^* and from Lemma 3.6.

By equation (3.20) and item 1 of Theorem 3.2, we obtain

$$\frac{d\|x(\gamma^*)\|^2}{d\gamma} = \frac{d\varphi(\lambda(\gamma^*) + \bar{\lambda})}{d\lambda} \frac{d\lambda(\gamma^*)}{d\gamma} \geq 0. \tag{3.31}$$

If $g''(\gamma^*) < 0$, then $g'(\gamma^* + h) < g'(\gamma^*) = 0$, for $h > 0$ small enough, and, using Lemma 3.6, we deduce $\|x(\gamma^* + h)\| < 1$. Thus, since $\|x(\gamma^*)\| = 1$,

$$\frac{d\|x(\gamma^*)\|^2}{d\gamma} = \lim_{h \to 0} \frac{\|x(\gamma^* + h)\|^2 - \|x(\gamma^*)\|^2}{h} \leq 0. \tag{3.32}$$

Inequalities (3.31) and (3.32) give

$$\frac{d\|x(\gamma^*)\|^2}{d\gamma} = 0.$$

It follows then from equality (3.31), and $\lambda'(\gamma^*) < 0$ that $\varphi'(\lambda(\gamma^*) + \bar{\lambda}) = 0$. From equation (3.6), φ is strictly convex over the interval $(\lambda_1(A), \lambda_2(A))$ and thus $\lambda(\gamma^*) + \bar{\lambda}$ is its strict minimizer. By equation (3.19), the following inequality thus holds

$$\|x(\gamma)\| \geq \|x(\gamma^*)\| = 1 \text{ for } \gamma \in \Gamma.$$

This contradicts $\|x(\gamma^* + h)\| < 1$ for $h > 0$ small enough. Thus $g''(\gamma^*) \geq 0$. □

Theorem 3.5. *Suppose $\gamma^* \in \mathbb{R}$ satisfies $g'(\gamma^*) = 0$ and $g''(\gamma^*) > 0$, then $x(\gamma^*)$ is a strict local non-global minimizer of* (2.2) *with Lagrange multiplier $\lambda^* := \lambda(\gamma^*) + \bar{\lambda}$. In addition, if $\lambda^* < 0$, then $x(\gamma^*)$ is a strict local-nonglobal minimizer of* (2.1).

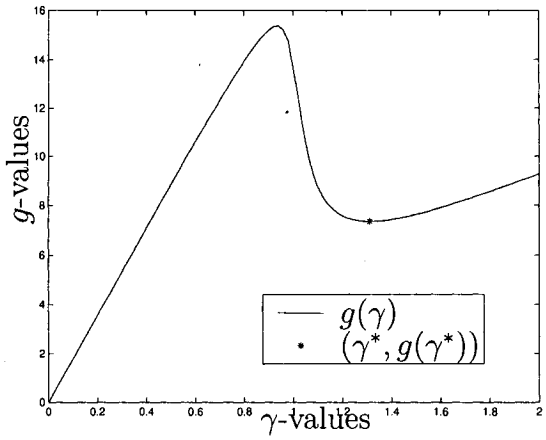

Figure 3.2: This figure illustrates Theorem 3.4.

Proof. From Lemma 3.3, $g''(\gamma^*) \neq 0$ implies $\lambda(\gamma^*) \in (\lambda_1(B), \lambda_2(B))$ and $\gamma^* \in \Gamma$. Therefore $x(\gamma^*)$ is well defined and if we let $x^* := x(\gamma^*)$ and $\lambda^* := \lambda(\gamma^*) + \bar{\lambda}$, then by equation (3.18), the stationarity condition (3.1) is satisfied. Feasibility of x^* follows from Lemma 3.6. If we can further show $\varphi'(\lambda^*) < 0$, then the result follows from item 3 of Theorem 3.2.

Since $g''(\gamma^*) > 0$, then $g'(\gamma^* - h) < g'(\gamma^*) = 0$ for $h > 0$ small enough. By Lemma 3.6, this implies $\|x(\gamma^* - h)\| < 1$ and thus

$$\frac{d\|x(\gamma^*)\|^2}{d\gamma} = \lim_{h \to 0} \frac{\|x(\gamma^*)\|^2 - \|x(\gamma^* - h)\|^2}{h} \geq 0. \tag{3.33}$$

By a similar argument that appears in the proof of Theorem 3.4, we conclude that the inequality in (3.33) holds strictly. From inequality (3.14) and equation (3.20), we deduce $\varphi'(\lambda^*) < 0$. □

Bounds on λ^* and γ^*

For this section, we assume a local-nonglobal minimizer of problem (2.2) exists. Our algorithm is based on finding a root γ^* to $\|x(\gamma)\|^2 - 1$ and we need initial bounds on γ^*. If $\mathcal{K} \cap \mathcal{J} = \varnothing$, then Lemma 3.3 implies $1 - d(\lambda_2(B)) < \gamma^* < \infty$. However, this

lower bound is of no practical utility, since we aim for an algorithm which exploits the sparsity of A and computing $d(\lambda_2(B))$ requires a full spectral decomposition of A. Otherwise, if $\mathcal{K} \cap \mathcal{J} \neq \varnothing$, the lemma does not gives us any supplementary information on bounds for γ^*, i.e. we only know $\gamma^* \in \mathbb{R}$. Our next lemma shows better bounds on γ^* exists, and these will improve the bounds on λ^* in Lemma 3.1 (Lemma 3.3 in Martínez [29]).

Lemma 3.7. *If a local-nonglobal minimizer of problems* (2.1) *or* (2.2) *exists, let* γ^* *be defined as in Theorem 3.3. Then* $\gamma^* \in [0, 2]$.

Proof. From Theorem 3.3, $x(\gamma^*)$ is the local-nonglobal minimizer and by feasibility

$$\|x(\gamma^*)\|^2 = \left(\frac{1-\gamma^*}{a^T v}\right)^2 v^T B^2 v = 1.$$

Therefore,

$$(1-\gamma^*)^2 = \frac{(a^T v)^2}{v^T B^2 v} \leq \left(\frac{\|a\|}{\lambda_1(B)}\right)^2,$$

where the last inequality follows from the Cauchy-Schwartz inequality and the fact that $v^T B^2 v \geq \lambda_1(B)^2 \|v\|^2$. Taking squares roots on both sides of the previous equation yields

$$\gamma^* \in \left[1 - \frac{\|a\|}{\lambda_1(B)}, 1 + \frac{\|a\|}{\lambda_1(B)}\right].$$

Finally, note from the definition of B that $\lambda_1(B) = \|a\|$. □

Corollary 3.3. *If a local-nonglobal minimizer of problems* (2.1) *or* (2.2) *exists, then* $\lambda^* \in [\bar{\lambda} + \lambda(2), \bar{\lambda} + \lambda(0)]$.

Proof. Recall, from Lemma 3.3, $\lambda(\gamma)$ is a decreasing function. □

Note

$$\bar{\lambda} + \lambda(2) > \bar{\lambda} + \lambda_1(B) = \lambda_1(A).$$

Thus the lower bound of Lemma 3.1 is improved and the upper bound is improved when $\lambda(0) < \lambda_2(B)$. It is also possible to deduce bounds on λ^* from the feasibility of x^*. Since $\varphi(\lambda^*) = 1$, then

$$\frac{\bar{a}_i^2}{(\lambda_i(A) - \lambda^*)^2} \leq 1 \text{ for all } i = 1 \ldots n.$$

Hence, by taking square roots on both sides

$$\lambda_1(A) + |\bar{a}_1| \leq \lambda^* \leq \min\{\lambda_i(A) - |\bar{a}_i| : i = 2 \ldots n\}.$$

The Algorithm

We are now ready to present an algorithm for either computing a possible local-nonglobal minimizer of problem (2.2) or either returning as an output that such a candidate does not exist. We will also indicate at the end of the section how this algorithm may be modified to compute a local-nonglobal minimizer of problem (2.1).

The upcoming algorithm is mainly the secant method. It exploits the fact that $\|x(\gamma)\|^2$ is strictly convex for $\gamma \in \Gamma$ and that we have an upper bound on γ^* when a local-nonglobal minimizer of problem (2.2) exists. To simplify our analysis, let $h(\gamma) := \|x(\gamma)\|^2 - 1$ and recall we are looking for a root of this function.

Algorithm 3.2.1.

1. INITIALIZATION
   - 1.1. Let $\gamma_L = 0$, $\gamma_U = 2$, $\gamma_0 = 2.1$, $\gamma_1 = \gamma_U$ and $k = 1$.
   - 1.2. If $\lambda(\gamma_U) = \lambda_2(B)$ or if $\frac{h(\gamma_1) - h(\gamma_0)}{\gamma_1 - \gamma_0} \leq 0$, LNGM $= 0$, else LNGM $= 1$.
2. ITERATION While $LNGM = 1$ and $\|x(\gamma_k)\| \neq 1$, do
   - 2.1. $\gamma_{k+1} = \gamma_k - \frac{h(\gamma_k)(\gamma_k - \gamma_{k-1})}{h(\gamma_k) - h(\gamma_{k-1})}$.
   - 2.2. If either $\lambda(\gamma_{k+1}) = \lambda_2(B)$, $\frac{h(\gamma_{k+1}) - h(\gamma_k)}{\gamma_{k+1} - \gamma_k} \leq 0$ or $\gamma_{k+1} < \gamma_L$, then LNGM $= 0$.
   - 2.3. $k = k + 1$.

The convergence results for Algorithm 3.2.1, which we are about to present, are based on the fact that we are using the secant method to find the root of a strictly convex function. To facilitate our analysis, we define the following linear function of γ which depends on the parameters γ_k and γ_{k-1}.

$$s(\gamma;\gamma_k,\gamma_{k-1}) := h(\gamma_k) + \frac{(h(\gamma_k) - h(\gamma_{k-1}))(\gamma - \gamma_k)}{\gamma_k - \gamma_{k-1}}.$$

The following lemma is a well known consequence of strict convexity for the function $h(\gamma)$.

Lemma 3.8. *Let $\gamma_k < \gamma_{k-1}$. For $\gamma \in \mathbb{R}$, the following inequalities and equality hold:*

1. $h(\gamma) < s(\gamma;\gamma_k,\gamma_{k-1})$ *if* $\gamma \in (\gamma_k,\gamma_{k-1})$,
2. $h(\gamma) = s(\gamma;\gamma_k,\gamma_{k-1})$ *if* $\gamma \in \{\gamma_k,\gamma_{k-1}\}$,
3. $h(\gamma) > s(\gamma;\gamma_k,\gamma_{k-1})$ *if* $\gamma \notin [\gamma_k,\gamma_{k-1}]$. □

In Algorithm 3.2.1, the secant iteration is initiated at γ_0 and γ_1 and halted if, for $k \geq 1$, $\lambda(\gamma_k) = \lambda_2(B)$ or the slope of the secant line going through the points $(\gamma_k, h(\gamma_k))$ and $(\gamma_{k-1}, h(\gamma_{k-1}))$ is not strictly positive. The next lemma shows, in the case these situations do not occur, that the sequence $\{\gamma_k\}$ produced by the secant iteration is strictly decreasing and converges to a root of h if bounded below. Such a bound could be γ_L for example.

Lemma 3.9. *Let γ_0 and γ_1 be defined as in Algorithm 3.2.1, and assume*

1. $s(\gamma_{k+1};\gamma_k,\gamma_{k-1}) = 0$, *for* $k \geq 1$,
2. $\gamma_k \in \Gamma$, *for* $k \geq 0$,
3. $\frac{h(\gamma_k)-h(\gamma_{k-1})}{\gamma_k-\gamma_{k-1}} > 0$, *for* $k \geq 1$.
4. $h(\gamma_1) > 0$.

Then $\{\gamma_k\}$ *is a strictly decreasing sequence. Furthermore, if* $\{\gamma_k\}$ *is bounded below, then the sequence converges to* $\bar{\gamma}$ *which satisfies* $h(\bar{\gamma}) = 0$ *and* $h'(\bar{\gamma}) \geq 0$.

Proof. Since $s(\gamma; \gamma_1, \gamma_0)$ is a function with positive slope by assumption, since γ_2 is its root and since $h(\gamma_1) > 0$, then clearly $\gamma_2 < \gamma_1$. By item 3 of Lemma 3.8, $h(\gamma_2) > s(\gamma_2; \gamma_1, \gamma_0) = 0$. By induction we may similarly prove $\{\gamma_k\}$ is a strictly decreasing sequence and $h(\gamma_k) > 0$ for $k \geq 0$.

If $\{\gamma_k\}$ is bounded below, because it is a decreasing sequence it converges, say to $\bar{\gamma}$. Now by the mean value theorem, for $k \geq 1$,

$$\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}} = h'(c_k) \quad \text{for } c_k \in [\gamma_k, \gamma_{k-1}].$$

By assumption $h'(c_k) > 0$, for $k \geq 1$, and since h is strictly convex, we deduce that for $k \geq 1$

$$\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}} < h'(c_0). \tag{3.34}$$

Convergence of $\{\gamma_k\}$ implies

$$0 = \lim_{k \to \infty} |\gamma_{k+1} - \gamma_k| = \lim_{k \to \infty} \frac{h(\gamma_k)}{\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}}}.$$

By (3.34) the denominator in the last limit is bounded away from infinity, hence the numerator converges to zero, i.e. $h(\bar{\gamma}) = 0$. Finally, since $c_k \in [\gamma_k, \gamma_{k-1}]$, then c_k converges to $\bar{\gamma}$ and thus $h'(\bar{\gamma}) = \lim_{k \to \infty} h'(c_k) \geq 0$. □

For our convergence results, we need to make one further assumption concerning problem (2.2).

Assumption 3.2.3. If problem (2.2) does not have a local-nonglobal minimizer, then for $\epsilon > 0$ small enough, the equality constrained trust-region subproblem

$$\begin{aligned} \min_x \quad & x^T A x - 2a^T x \\ \text{s.t.} \quad & \|x\| = 1 + \epsilon. \end{aligned} \tag{3.35}$$

does not have a local-nonglobal minimizer.

The scalar 1 on the right hand side of the equality constraint of problem (3.35) is due the fact we assume $\Delta = 1$ in problem (2.2). Now consider for a moment that Δ is allowed to vary in (2.2). Our assumption mainly says that for Δ larger but close enough to 1, there are no local-nonglobal minimizer. Note if Assumption 3.2.3 holds, that for $\Delta < 1$ there are no local-nonglobal minimizer as well. This is a consequence of Theorem 3.2 and of the strict convexity of the function φ define in (3.4). Furthermore, in view of Lemma 3.10, there exists Δ_0 such that (2.2) admits a local-nonglobal minimizer for all $\Delta > \Delta_0$. Thus in case a local-nonglobal minimizer does not exist for (2.2) the assumption is equivalent to the inequality $1 < \Delta_0$.

Under the extra Assumption 3.2.3, the following theorem holds.

Theorem 3.6. *The sequence $\{\gamma_k\}$ produced by Algorithm 3.2.1 either converges to γ^* such that $x(\gamma^*)$ is a local-nonglobal minimizer of problem* (2.2) *or there does not exist a local-nonglobal minimizer of problem* (2.2) *and LNGM is set to 0.*

Proof. First, consider the case where a local-nonglobal minimizer for problem (2.2) exists. Let γ^* be defined as in Theorem 3.3. Then

$$h(\gamma^*) = 0 \qquad \text{and} \qquad h'(\gamma^*) \geq 0. \tag{3.36}$$

Recall γ_1 is an upper bound on γ^*. If $\gamma_1 = \gamma^*$, then $\|x(\gamma_1)\| = 1$. Hence, Algorithm 3.2.1 terminates and there is nothing to prove.

Assume $\gamma_1 > \gamma^*$. Note, from Lemma 3.1, $\lambda^* \in (\lambda_1(A), \lambda_2(A))$ and from Theorem 3.3, $\lambda^* = \lambda(\gamma^*) + \bar{\lambda}$. Hence $\lambda(\gamma^*) \in (\lambda_1(B), \lambda_2(B))$, i.e. $\gamma^* \in \Gamma$. Since $\lambda(\gamma)$ is a decreasing function, $\lambda(\gamma_1) \leq \lambda(\gamma^*) < \lambda_2(B)$. From Corollary 3.2, $\lambda(\gamma_1) > \lambda_1(B)$. Hence, $\gamma_1 \in \Gamma$. Since h is strictly convex, since $\gamma_1 > \gamma^*$ and since (3.36) holds, $h(\gamma_1) > 0$ and $h'(\gamma_1) > 0$.

Suppose for $k \geq 1$ that $\gamma_k > \gamma^*$, $\gamma_k \in \Gamma$, $h(\gamma_k) > 0$ and $h'(\gamma_k) > 0$. Note we just proved these conditions hold for $k = 1$. We wish to show

1. $\frac{h(\gamma_k)-h(\gamma_{k-1})}{\gamma_k-\gamma_{k-1}} > 0$,
2. $h(\gamma_{k+1}) > 0$,
3. $\gamma_{k+1} > \gamma^*$,
4. $h'(\gamma_{k+1}) > 0$,
5. $\gamma_{k+1} \in \Gamma$.

Since

$$\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}} = h'(c_k) \quad \text{for } c_k \in [\gamma_k, \gamma_{k-1}],$$

and h is strictly convex, then $h'(c_k) \geq h'(\gamma_k) > 0$, proving item 1. It follows $s(\gamma; \gamma_k, \gamma_{k-1})$ is strictly increasing. Since $0 = s(\gamma_{k+1}; \gamma_k, \gamma_{k-1})$ and $s(\gamma_k; \gamma_k, \gamma_{k-1}) = h(\gamma_k) > 0$, then $\gamma_{k+1} < \gamma_k$. By Lemma 3.8, $h(\gamma_{k+1}) > s(\gamma_{k+1}; \gamma_k, \gamma_{k-1}) = 0$ and this proves item 2. We have $\gamma^* < \gamma_{k+1}$, otherwise $\gamma^* \in (\gamma_{k+1}, \gamma_k)$ and by Lemma 3.8

$$0 = s(\gamma_{k+1}; \gamma_k, \gamma_{k-1}) < s(\gamma^*; \gamma_k, \gamma_{k-1}) < h(\gamma^*).$$

This contradicts $h(\gamma^*) = 0$, proving item 3. Since $h'(\gamma^*) \geq 0$, by strict convexity we have $h'(\gamma_{k+1}) > 0$, proving item 4. Finally, from an argument similar as above, $\gamma_{k+1} \in \Gamma$ holds, proving item 5.

By induction it follows that for all $k \geq 1$

1. $\frac{h(\gamma_k)-h(\gamma_{k-1})}{\gamma_k-\gamma_{k-1}} > 0$,
2. $h(\gamma_k) > 0$,
3. $\gamma_k > \gamma^*$,
4. $h'(\gamma_k) > 0$,
5. $\gamma_k \in \Gamma$.

It follows from Lemma 3.9 that $\{\gamma_k\}$ converges, say to $\bar{\gamma}$, which satisfies $h(\bar{\gamma}) = 0$ and $h'(\bar{\gamma}) \geq 0$. By strict convexity of h and since $h(\gamma^*) = 0$ and $h'(\gamma^*) \geq 0$, then $\bar{\gamma} = \gamma^*$.

Second, consider the case where a local-nonglobal minimizer of problem (2.2) does not exist. Suppose there exists $\hat{\gamma} \in \Gamma$ such that $h(\hat{\gamma}) < 0$, then since h is strictly convex and, by Lemma 3.5, since $\lim_{\gamma \to \infty} h(\gamma) = \infty$, there exists $\gamma^* > \hat{\gamma}$ such that $\gamma^* \in \Gamma$, $h(\gamma^*) = 0$ and $h'(\gamma^*) > 0$. By Theorem 3.3, there exists then a local-nonglobal minimizer of problem (2.2), yielding a contradiction. Thus, $h(\gamma) \geq 0$ for $\gamma \in \Gamma$. In fact, this inequality holds strictly. Otherwise, from the strict convexity of h, for every $\epsilon > 0$, the equation $h(\gamma) = \epsilon(\epsilon + 2)$ has a solution, say $\hat{\gamma} \in \Gamma$, with $h'(\hat{\gamma}) > 0$. From item 3 of Theorem 3.3, $x(\hat{\gamma})$ is a local-nonglobal minimizer of problem (3.21) with $\|x(\hat{\gamma})\| = 1 + \epsilon$. This contradicts Assumption 3.2.3. Hence $h(\gamma) > 0$ for $\gamma \in \Gamma$. We obtain in particular that $h(\gamma_1) > 0$.

If a sequence $\{\gamma_k\}$ obtained with Algorithm 3.2.1 would be bounded below and satisfy, for all $k \geq 1$, $\gamma_k \in \Gamma$ and

$$\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}} > 0,$$

then by Lemma 3.9, $\{\gamma_k\}$ would converge, say to $\bar{\gamma}$, which would satisfy $h(\bar{\gamma}) = 0$ and $h'(\bar{\gamma}) \geq 0$. This would imply that for every $\epsilon > 0$, the equation $h(\gamma) = \epsilon(\epsilon + 2)$ has a solution, say $\hat{\gamma} \in \Gamma$, with $h'(\hat{\gamma}) > 0$, contradicting as explained above Assumption 3.2.3. Thus, there must exist some $\bar{k} \geq 1$ such that either one of the following cases is true:

1. $\gamma_{\bar{k}} < \gamma_L$,
2. $\frac{h(\gamma_{\bar{k}}) - h(\gamma_{\bar{k}-1})}{\gamma_{\bar{k}} - \gamma_{\bar{k}-1}} \leq 0$,
3. $\gamma_{\bar{k}} \notin \Gamma$ (if and only if $\lambda(\gamma_{\bar{k}}) = \lambda_2(B)$).

In either cases, a local non-global minimizer of problem (2.2) does not exist and our algorithm sets LNGM = 0. □

Corollary 3.4. *Suppose x^* is a local-nonglobal minimizer of problem (2.2) with a corresponding Lagrange multiplier λ^* that satisfies (3.1). Let γ^* be the unique solution to $\lambda(\gamma) + \bar{\lambda} = \lambda^*$. If $h'(\gamma^*) > 0$, the sequence $\{\gamma_k\}$ produced by Algorithm 3.2.1 converges to γ^* superlinearly and $x(\gamma^*)$ is a strict local-nonglobal minimizer of problem (2.2).*

Proof. The proof holds from Theorem 3.3, Theorem 3.6 and since the secant method converges superlinearly when it converges to a simple root, see e.g. Neumaier [33, Corollary 5.4.2]. □

We now discuss how we can modifies Algorithms 3.2.1 in order to compute a local-nonglobal minimizer of problem (2.1).

From item 2 of Theorem 3.1, if x^* is a local-nonglobal minimizer of problem (2.1), then $\|x^*\| = 1$ must hold. Hence x^* is necessarily a local-nonglobal minimizer of problem (2.2). From the standard necessary optimality conditions it has a negative Lagrange multiplier $\lambda^* \leq 0$. Furthermore, it is shown in [27] that in fact strict inequality holds, i.e. $\lambda^* < 0$.

Recall that as long as LNGM = 1 the sequence $\{\gamma_k\}$ generated by Algorithm 3.2.1 is decreasing and that the function $\lambda(\gamma) + \bar{\lambda}$ is decreasing. Therefore, in Algorithm 3.2.1, if at some iteration k,

$$\lambda(\gamma_k) + \bar{\lambda} \geq 0, \tag{3.37}$$

we may deduce from Theorem 3.3 that problem (2.1) does not have a local-nonglobal minimizer. Thus in order to modify Algorithm 3.2.1 for computing a local-nonglobal minimizer of problem (2.1), we may set the boolean parameter LNGM to 0 whenever (3.37) holds. With this change to the algorithm, Theorem 3.6 and Corollary 3.4 also hold if problem (2.2) is replaced by problem (2.1) in the statements.

Local Optimums of TRS for Infinitely Large Trust Regions

Consider problem (2.2) where Δ is now any positive number. We wish to investigate the limiting behavior of local (global and nonglobal) minimizers of problem (2.2) as $\Delta \to \infty$. Obviously, x^* is a local minimizer of problem (2.2) if and only if x^*/Δ is a local minimizer of the following problem:

$$\begin{aligned} \min_x \quad & x^T(\Delta^2 A)x - 2(\Delta a)^T x \\ \text{s.t.} \quad & \|x\| = 1. \end{aligned} \tag{3.38}$$

Thus equivalently, we investigate the limiting behavior of local minimizers of problem (3.38) as $\Delta \to \infty$.

In this new setting, redefine $\bar{\lambda} := \Delta^2\lambda_1(A) - \Delta\|a\|$ and $B := \Delta^2 A - \bar{\lambda}I$. Recall $A = QDQ^T$. Let $Q = [q_1, \ldots, q_n]$, where q_i is the i-th column of Q.

If we define $x_G(\Delta)$ and $x_L(\Delta)$ to be respectively the global and the local-nonglobal minimizers of (3.38), then Martínez [29] has shown the following

Lemma 3.10. *There exists $\Delta_0 > 0$ such that* (3.38) *admits a local-nonglobal minimizer for all $\Delta > \Delta_0$ and*

$$x_G(\infty) := \lim_{\Delta\to\infty} x_G(\Delta) = \frac{\bar{a}_1}{|\bar{a}_1|}q_1, \tag{3.39}$$

$$x_L(\infty) := \lim_{\Delta\to\infty} x_L(\Delta) = -\frac{\bar{a}_1}{|\bar{a}_1|}q_1. \tag{3.40}$$

Similarly to how problem (2.8) was derived, we rewrite (3.38) as

$$\begin{aligned} \bar{\lambda} - \Delta^2 a^T B^{-1} a + \min_x \quad & r_\Delta^2(x) := (x - \Delta B^{-1}a)^T B(x - \Delta B^{-1}a) \\ \text{s.t.} \quad & \|x\| = 1. \end{aligned} \tag{3.41}$$

Our goal is to show equations (3.39) and (3.40) may be deduced from the geometry of problem (3.41). Define for $p > 0$ the level set

$$\Omega_{p,\Delta} := \{x : (x - B^{-1}\Delta a)^T B(x - B^{-1}\Delta a) \le p^2\lambda_1(B)\}.$$

This set is an ellipsoid centered in the interior of the unit ball and bounded by the level curve $r_\Delta^2(x) = p^2\lambda_1(B)$. The choice of the latter constant is made to simplify the upcoming expressions. It follows x^* is a local minimizer of problem (3.38) if and only if, for some $p > 0$, $x^* \in \Omega_{p,\Delta} \cap \{x : \|x\| = 1\}$ and, for some $\delta > 0$, $\Omega_{p,\Delta} \cap \{x : \|x - x^*\| \leq \delta, \|x\| > 1\} = \varnothing$. This means x^* lies on the boundary of $\Omega_{p,\Delta}$, for some $p > 0$, which is locally contained at x^* in the unit ball and tangent to the unit sphere at x^*.

We have $B = Q(\Delta^2 D - \bar{\lambda} I)Q^T$, so that $B = QD_BQ^T$, where

$$D_B := \begin{pmatrix} \Delta\|a\| & & & 0 \\ & \Delta^2(\lambda_2(A) - \lambda_1(A)) + \Delta\|a\| & & \\ & & \ddots & \\ 0 & & & \Delta^2(\lambda_n(A) - \lambda_1(A)) + \Delta\|a\| \end{pmatrix}.$$

Note $\lambda_i(B) = (D_B)_{ii}$ for $i = 1 \ldots n$. Now we may write $\Omega_{p,\Delta}$ as

$$\Omega_{p,\Delta} := \{x : (Q^T(x - B^{-1}\Delta a))^T D_B (Q^T(x - B^{-1}\Delta a)) \leq p^2\lambda_1(B)\},$$

It follows

$$\Omega_{p,\Delta} = \left\{ x = Qz + \Delta B^{-1}a : \frac{z_1^2}{p^2} + \frac{z_2^2}{\left(\frac{p^2\lambda_1(B)}{\lambda_2(B)}\right)} + \ldots + \frac{z_n^2}{\left(\frac{p^2\lambda_1(B)}{\lambda_n(B)}\right)} \leq 1 \right\}. \tag{3.42}$$

Now, for $i = 2 \ldots n$,

$$\lim_{\Delta\to\infty} \frac{\lambda_1(B)}{\lambda_i(B)} = \lim_{\Delta\to\infty} \frac{\|a\|}{\|a\| + \Delta(\lambda_i(A) - \lambda_1(A))} = 0$$

and

$$\begin{aligned} \lim_{\Delta\to\infty} \Delta B^{-1}a &= \lim_{\Delta\to\infty} Q(\Delta D_B^{-1})\bar{a}, \\ &= \lim_{\Delta\to\infty} \frac{\bar{a}_1}{\|a\|}q_1 + \frac{\bar{a}_2}{\Delta(\lambda_2(A) - \lambda_1(A)) + \|a\|}q_2 + \ldots \\ &+ \frac{\bar{a}_n}{\Delta(\lambda_n(A) - \lambda_1(A)) + \|a\|}q_n, \\ &= \frac{\bar{a}_1}{\|a\|}q_1, \end{aligned} \tag{3.43}$$

where the last equality follows by Assumption 3.2.1. Hence, as Δ becomes large, the length of the $n-1$ smaller axis of the ellipsoid (3.42) tend to zero, the length of the larger axis tends to $2p$ and the center of the ellipsoid tends to $\frac{\bar{a}_1}{\|a\|}q_1$. In other words, as Δ becomes large, the ellipsoid (3.42) converges to the segment

$$\Omega_{p,\infty} := \left\{ x = Qz + \frac{\bar{a}_1}{\|a\|}q_1 : |z_1| \leq p \, ; z_i = 0 \text{ for } i = 2 \ldots n \right\},$$

which can be rewritten as

$$\Omega_{p,\infty} = \left\{ x = \left(z_1 + \frac{\bar{a}_1}{\|a\|} \right) q_1 : |z_1| \leq p \right\}. \tag{3.44}$$

Therefore, $x_G(\infty)$ and $x_L(\infty)$ are obtained from the intersection of the boundary of the unit sphere with an end point of a segment of the form $\Omega_{p,\infty}$. We have to look for values of $p = |\alpha|$ such that

$$|\alpha + \bar{a}_1/\|a\|| = 1. \tag{3.45}$$

There are two values of α that satisfy (3.45):

$$\alpha_1 := 1 - \bar{a}_1/\|a\| \text{ and } \alpha_2 := -1 - \bar{a}_1/\|a\|.$$

Now let

$$m := \min\{|\alpha_1|, |\alpha_2|\} = 1 - |\bar{a}_1|/\|a\|.$$

Therefore $x_G(\infty)$ is the intersection of the limiting level set $\Omega_{m,\infty}$ with the unit sphere. It follows

$$x_G(\infty) = \begin{cases} q_1 & \text{if } \bar{a}_1 \geq 0 \\ -q_1 & \text{if } \bar{a}_1 < 0. \end{cases}$$

Thus (3.39) holds. Similarly, equation (3.40) holds, since for

$$M := \max\{|\alpha_1|, |\alpha_2|\} = 1 + |\bar{a}_1|/\|a\|,$$

we have that $x_L(\infty)$ is the intersection of the end points of the limiting level set $\Omega_{M,\infty}$ with the unit sphere.

For each value of Δ, there exists a value of p for which the global minimizer $x_G(\Delta)$ lies in the intersection of the unit sphere and an ellipsoid $\Omega_{p,\Delta}$ contained in the unit sphere. In Figure 3.3, on the left side, we illustrate, as Δ varies, this sequence of ellipsoids $\Omega_{p,\Delta}$. One sees, as Δ becomes large, that the ellipsoids converge to the segment $\Omega_{m,\infty}$. Figure 3.3, on the right side, illustrates the same concept, but this time, for different values of Δ, we plot the ellipsoids $\Omega_{p,\Delta}$ locally contained at the local-nonglobal minimizers $x_L(\Delta)$. In this case, the ellipsoids converge to the segment $\Omega_{M,\infty}$.

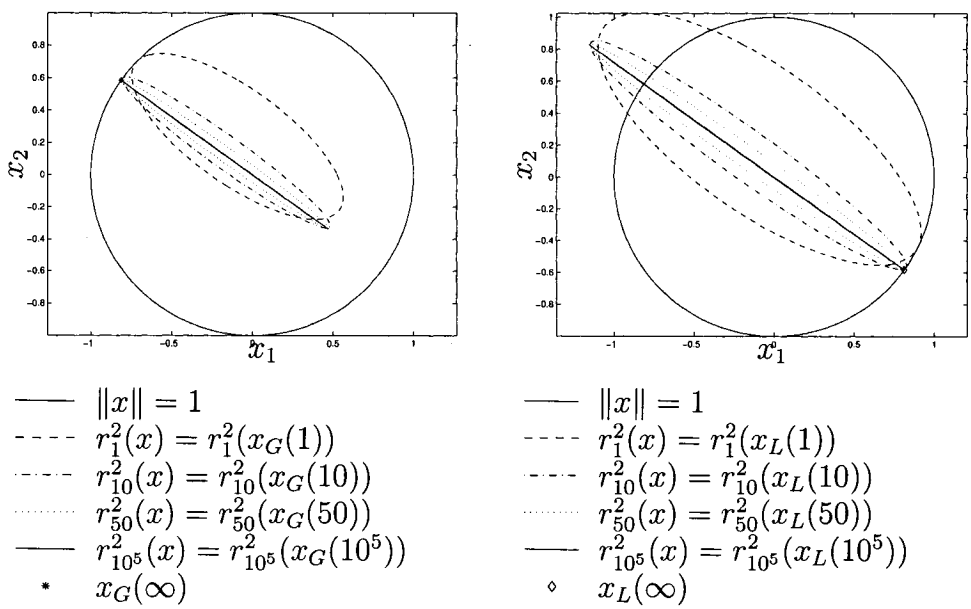

Figure 3.3: Limiting behavior as $\Delta \to \infty$ of the ellipsoids $\Omega_{m,\Delta}$ to which the points $x_G(\Delta)$ (on the right) and $x_L(\Delta)$ (on the left) are associated.

3.2.2 Computing a Local-Nonglobal Minimizer: Second Method

Rendl and Wolkowicz [39] have proposed an algorithm for computing the global minimizer of problem (2.2). The algorithm is based on the problem

$$\max_{t\in\mathbb{R}} \quad k(t) := 2\lambda_1(D(t)) - t, \tag{3.46}$$

where

$$D(t) := \begin{bmatrix} t & -a^T \\ -a & A \end{bmatrix}.$$

Its optimal objective value is the same as the one of problem (2.2). Problem (3.46) is useful since k is a concave function over $\mathbb{R}$ and if λ^* is the Lagrange multiplier of a global optimal solution x^* for problem (2.2), then $\lambda^* = \lambda_1(D(t^*))$, where t^* is optimal for problem (3.46).

The matrix $D(t)$ is of dimension $n+1$ and is simply the matrix A to which a new row and column was added. The fact that the eigenvalues of A interlace those of $D(t)$ is known as Cauchy's inequalities [5] or the interlacing eigenvalues theorem for bordered matrices. It is formally stated in the next lemma and a proof may be found in [23].

Lemma 3.11. *For $t \in \mathbb{R}$,*

$$\lambda_1(D(t)) \leq \lambda_1(A) \leq \lambda_2(D(t)) \leq \lambda_2(A) \leq \ldots \leq \lambda_n(D(t)) \leq \lambda_n(A) \leq \lambda_{n+1}(D(t)). \quad \square$$

Our next lemma studies the properties of the function $\lambda_2(D(t))$. Our interest in this function comes from the fact that its image lies in the interval $[\lambda_1(A), \lambda_2(A)]$, the same interval where lies the Lagrange multiplier of a local-nonglobal minimizer of problem (2.2). We will make use in our analysis of the function

$$p(\lambda) := \lambda + \sum_{j\in\mathcal{J}} \frac{\bar{a}_j^2}{\lambda_j(A) - \lambda}$$

and of the sets $\mathcal{J}$ and $\mathcal{K}$ defined in (3.7).

Lemma 3.12. *1. If $\mathcal{K} \cap \mathcal{J} \neq \varnothing$, $\lambda_2(D(t))$ is infinitely differentiable and satisfies $p(\lambda_2(D(t))) = t$. Moreover,*

$$\lambda_2'(D(t)) = \frac{1}{p'(\lambda_2(D(t)))} \text{ and } \lambda_2''(D(t)) = \frac{-p''(\lambda_2(D(t)))}{[p'(\lambda_2(D(t)))]^3} \tag{3.47}$$

for all $t \in \mathbb{R}$.

2. If $\mathcal{K} \cap \mathcal{J} = \varnothing$, $\lambda_2(D(t))$ is continuous and infinitely differentiable for $t \in \mathbb{R} \setminus \{p(\lambda_2(A))\}$.

(a) For $t < p(\lambda_2(A))$, $\lambda_2(D(t))$ satisfies $p(\lambda_2(D(t))) = t$ and

$$\lambda_2'(D(t)) = \frac{1}{p'(\lambda_2(D(t)))} \text{ and } \lambda_2''(D(t)) = \frac{-p''(\lambda_2(D(t)))}{[p'(\lambda_2(D(t)))]^3}. \tag{3.48}$$

(b) For $t > p(\lambda_2(A))$, $\lambda_2(D(t)) = \lambda_2(A)$, $\lambda_2'(D(t)) = 0$ and $\lambda_2''(D(t)) = 0$.

(c) for $t = p(\lambda_2(A))$, $\lambda_2(D(t)) = \lambda_2(A)$ and satisfies $p(\lambda_2(D(t))) = t$; the right handed and left handed derivatives are given respectively by

$$\lambda_2'(D(t^+)) = \frac{1}{p'(\lambda_2(A))}, \quad \lambda_2''(D(t^+)) = \frac{-p''(\lambda_2(A))}{[p'(\lambda_2(A))]^3}, \tag{3.49a}$$

$$\lambda_2'(D(t^-)) = 0, \quad \lambda_2''(D(t^-)) = 0. \tag{3.49b}$$

Proof. 1. We have

$$D(t) - \lambda I = \begin{bmatrix} 1 & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} t-\lambda & -\bar{a}^T \\ -\bar{a} & D - \lambda I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q \end{bmatrix}^T.$$

Expanding the determinant of the diagonal matrix with respect to its first column gives

$$\det(D(t) - \lambda I) = (t-\lambda)\prod_{j=1}^{n}(\lambda_j(A) - \lambda) - \sum_{k=1}^{n}\left(\bar{a}_k^2 \prod_{j\neq k}^{n}(\lambda_j(A) - \lambda)\right), \tag{3.50a}$$

$$= (t - p(\lambda))\prod_{j=1}^{n}(\lambda_j(A) - \lambda) \quad \text{for } \lambda \notin \{\lambda_j(A) | j \in J\}. \tag{3.50b}$$

Since $\mathcal{K} \cap \mathcal{J} \neq \varnothing$ and $\bar{a}_1 \neq 0$, then

$$\lim_{\lambda \searrow \lambda_1(A)} p(\lambda) = -\infty \quad \text{and} \quad \lim_{\lambda \nearrow \lambda_2(A)} p(\lambda) = \infty.$$

Furthermore,

$$p'(\lambda) = 1 + \sum_{j \in \mathcal{J}} \frac{\bar{a}_j^2}{(\lambda_j(A) - \lambda)^2} > 0.$$

Therefore, for all $t \in \mathbb{R}$, $p^{-1}(t)$ is well defined, where $p^{-1}(t) \in (\lambda_1(A), \lambda_2(A))$. Moreover, (3.50b) shows it is an eigenvalue of $D(t)$. From Lemma 3.11, this shows $p^{-1}(t) = \lambda_2(D(t))$. Hence, $\lambda_2(D(t))$ is infinitely differentiable and equations (3.47) are obtained by implicit differentiation.

2. Let $t \in \mathbb{R}$. Since $\mathcal{K} \cap \mathcal{J} = \varnothing$, then $p(\lambda_2(A))$ is well defined. By equation (3.50b), $\lambda_2(A)$ is an eigenvalue of $D(t)$. Assumptions 3.2.1 and 3.2.2 imply $\lambda_1(A)$ is not an eigenvalue of $D(t)$, since from equation (3.50a) we obtain

$$\det(D(t) - \lambda_1(A)I) = -\bar{a}_1^2 \prod_{j=2}^{n} (\lambda_j(A) - \lambda_1(A)) \neq 0.$$

Combining this last inequality with Lemma 3.11 gives

$$\lambda_2(D(t)) \in (\lambda_1(A), \lambda_2(A)]. \tag{3.51}$$

Note again $p(\lambda)$ is strictly increasing for $\lambda \in (\lambda_1(A), \lambda_2(A)]$ and therefore

$$p(\lambda) < p(\lambda_2(A)) \quad \text{for} \quad \lambda \in (\lambda_1(A), \lambda_2(A)). \tag{3.52}$$

(a) . If $t < p(\lambda_2(A))$, then $p^{-1}(t)$ is well defined, where $p^{-1}(t) \in (\lambda_1(A), \lambda_2(A))$. The rest of the proof is similar to the proof of item 1.

(b) and (c). If $t \geq p(\lambda_2(A))$, by (3.52), $p(\lambda) < t$ for $\lambda \in (\lambda_1(A), \lambda_2(A))$. Therefore, from equation (3.50b), there are no eigenvalues of $D(t)$ in the interval $[\lambda_1(A), \lambda_2(A))$ and, using the expression (3.51), we obtain $\lambda_2(A) = \lambda_2(D(t))$. In particular, the derivatives of $\lambda_2(D(t))$ for $t > p(\lambda_2(A))$ are

zero and equations (3.49b) hold. Note finally $t = p(\lambda_2(D(t)))$ for $t \leq p(\lambda_2(A))$ and thus (3.49a) holds.

□

Corollary 3.5. *For $t \in \mathbb{R}$, $\lambda_2(D(t)) > \lambda_1(A)$ and $\lim_{t\to-\infty} \lambda_2(D(t)) = \lambda_1(A)$. Moreover,*

1. *if $\mathcal{K} \cap \mathcal{J} \neq \varnothing$, $\lambda_2(D(t)) < \lambda_2(A)$ and $\lim_{t\to\infty} \lambda_2(D(t)) = \lambda_2(A)$.*
2. *If $\mathcal{K} \cap \mathcal{J} = \varnothing$,*
   (a) *$\lambda_2(D(t)) = \lambda_2(A)$ for $t \geq p(\lambda_2(A))$,*
   (b) *$\lambda_2(D(t)) < \lambda_2(A)$ for $t < p(\lambda_2(A))$.* □

Analogously to how function g was defined in Section §3.2.1, we define the function

$$m(t) := 2\lambda_2(D(t)) - t,$$

which has the real axis for domain. Furthermore, define

$$\mathcal{T} := \begin{cases} \mathbb{R}, & \text{if } \mathcal{K} \cap \mathcal{J} \neq \varnothing, \\ (-\infty, p(\lambda_2(A)), & \text{if } \mathcal{K} \cap \mathcal{J} = \varnothing. \end{cases}$$

Note, from Lemma 3.12,

$$\mathcal{T} = \{t : \lambda_2(D(t)) \in (\lambda_1(A), \lambda_2(A))\}, \tag{3.53}$$

$$\lambda_2'(D(t)) > 0 \quad \text{for} \quad t \in \mathcal{T}. \tag{3.54}$$

For $t \in \mathcal{T}$, let the vector $y \in \mathbb{R}^n$ be a unit norm eigenvector of $D(t)$ for the eigenvalue $\lambda_2(D(t))$, i.e.

$$D(t)y = \lambda_2(D(t))y. \tag{3.55}$$

Note y depends on t. Let $y := (y_0, z)^T$, where $y_0 \in \mathbb{R}$ and $z \in \mathbb{R}^n$. Thus, the derivative of m is

$$m'(t) = 2y_0^2 - 1. \tag{3.56}$$

When $\mathcal{K} \cap \mathcal{J} = \varnothing$ and $t > p(\lambda_2(A))$, then $m'(t) = -1$. For $t = p(\lambda_2(A))$, it may happen that m is not differentiable, since it is possible to show the multiplicity of $\lambda_2(D(t))$ is changing.

From equation (3.55), we have

$$ty_0 - a^T z = \lambda_2(D(t))y_0, \tag{3.57}$$

$$-y_0 a + Az = \lambda_2(D(t))z. \tag{3.58}$$

Now for $t \in \mathcal{T}$, define

$$x(t) := \frac{z}{y_0}. \tag{3.59}$$

Note

$$y_0 \neq 0 \quad \text{for } t \in \mathcal{T}, \tag{3.60}$$

otherwise, by equation (3.58), this would imply $\lambda_2(D(t))$ is an eigenvalue of A. Rewriting this same equation, we obtain that $x(t)$ satisfies the first order necessary condition (3.1), i.e.

$$(A - \lambda_2(D(t))I)x(t) = a. \tag{3.61}$$

For $t \in \mathcal{T}$, $\lambda_2(D(t)) \in (\lambda_1(A), \lambda_2(A))$. Therefore $A - \lambda_2(D(t))I$ is invertible, and we may write $\|x(t)\|^2$ as

$$\|x(t)\|^2 = \varphi(\lambda_2(D(t))). \tag{3.62}$$

It follows

$$\frac{d\|x(t)\|^2}{dt} = \frac{d\varphi(\lambda_2(D(t)))}{d\lambda}\frac{d\lambda_2(D(t))}{dt}. \tag{3.63}$$

Again, the algorithm of this section is based on finding a root of the function $\|x(t)\|^2 - 1$ in the interval $\mathcal{T}$. A theorem similar to Theorem 3.2 holds.

Theorem 3.7. *1. If x^* is a local-nonglobal minimizer of problems* (2.1) *or* (2.2), *then* (3.1) *holds with $\lambda^* \in (\lambda_1(A), \lambda_2(A))$. Let t^* be the unique solution to $\lambda_2(D(t)) = \lambda^*$, then $x^* = x(t^*)$ and $\frac{d\|x(t^*)\|^2}{dt} \leq 0$. If x^* is a local-nonglobal minimizer of* (2.1) *then $\lambda_2(D(t^*)) \leq 0$.*

2. *If, for* $t^* \in \mathcal{T}$, $\|x(t^*)\| = 1$ *and* $\frac{d\|x(t^*)\|^2}{dt} < 0$, *then* $x(t^*)$ *is a strict local-nonglobal minimizer of* (2.2). *If in addition,* $\lambda_2(D(t^*)) < 0$, $x(t^*)$ *is also a strict local-nonglobal minimizer of* (2.1).

3. *For* $t \in \mathcal{T} \cap \{t : \frac{d\|x(t)\|^2}{dt} < 0\}$, $x(t)$ *is a strict local non-global minimizer of*

$$\begin{aligned} \min_x \quad & x^T Ax - 2a^T x \\ s.t. \quad & \|x\| = \|x(t)\| \end{aligned} \tag{3.64}$$

with Lagrange multiplier $\lambda_2(D(t))$. *In addition, if* $\lambda_2(D(t)) < 0$, *then* $x(t)$ *is a strict local-nonglobal minimizer of*

$$\begin{aligned} \min_x \quad & x^T Ax - 2a^T x \\ s.t. \quad & \|x\| \leq \|x(t)\|. \end{aligned} \tag{3.65}$$

Proof. Since $t^* \in \mathcal{T}$, the proofs of items 1 and 2 follow from Theorem 3.2 and equations (3.54), (3.62) and (3.63). To prove item 3, fix $t \in \mathcal{T}$ and let $\delta := \|x(t)\|$. Note $x(t)$ is a local-nonglobal minimizer of problem (3.64) if and only if problem (3.23) is solved by $x(t;\delta) := x(t)/\delta$.

For $t \in \mathcal{T} \cap \{t : \frac{d\|x(t)\|^2}{dt} < 0\}$, it follows by equations (3.54) and (3.63) that

$$\frac{d\varphi(\lambda_2(D(t)))}{d\lambda} < 0. \tag{3.66}$$

By item 3 of Theorem 3.2, $\|x(t;\delta)\|$ solves problem (3.23) since

$$((\delta^2 A) - (\delta^2 \lambda_2(D(t)))I)x(t;\delta) = \delta a,$$

since $\lambda_1(\delta^2 A) < \delta^2 \lambda_2(D(t)) < \lambda_2(\delta^2 A)$ and since, from the equation (3.24) and the inequality (3.66),

$$\varphi'(\delta^2 \lambda_2(D(t));\delta) = \varphi'(\lambda_2(D(t))) < 0.$$

□

As for Algorithm 3.2.1, a key property for the algorithm of this section is that $\|x(t)\|^2$ is a strictly convex function.

Lemma 3.13. *Consider the function $\|x(t)\|^2$ with domain $\mathcal{T}$. Then it is an infinitely differentiable strictly convex function and $\lim_{t\to-\infty}\|x(t)\|^2 = \infty$.*

Proof. Since $\varphi(\lambda)$ and $\lambda_2(D(t))$ are infinitely differentiable respectively on the intervals $(\lambda_1(A), \lambda_2(A))$ and $\mathcal{T}$, and $\lambda_2(D(t)) \in (\lambda_1(A), \lambda_2(A))$, then infinite differentiability follows from (3.62). By Corollary 3.5, $\lim_{t\to-\infty}\lambda_2(D(t)) = \lambda_1(A)$ and $\lambda_2(D(t)) > \lambda_1(A)$, and, by Assumption 3.2.2, $\lim_{\lambda\searrow\lambda_1(A)}\varphi(\lambda) = \infty$. Thus, using equation (3.62), $\lim_{t\to-\infty}\|x(t)\|^2 = \infty$.

All that is left to prove is strict convexity. For simplicity, let $\lambda_i = \lambda_i(A)$, for $i = 1,\ldots,n$, let $\lambda_t = \lambda_2(D(t))$ and let $\lambda_t' = \lambda_2'(D(t))$. There are two cases to consider.

1. *Case 1: $\bar{a}_1 \neq 0$ and $\bar{a}_j = 0$ for $j = 2,\ldots,n$.* We have in this case

$$\begin{aligned}\|x(t)\|^2 &= \frac{\bar{a}_1^2}{(\lambda_1-\lambda_t)^2},\\ \frac{d\|x(t)\|^2}{dt} &= \frac{2\bar{a}_1^2}{(\lambda_1-\lambda_t)^3}\lambda_t' = \frac{2\bar{a}_1^2}{(\lambda_1-\lambda_t)^3+\bar{a}_1^2(\lambda_1-\lambda_t)},\end{aligned}$$

   where we have used equation (3.47) and (3.48) to obtain the first derivative. Thus, using equation (3.54),

$$\frac{d^2\|x(t)\|^2}{dt^2} = \frac{2\bar{a}_1^2(3(\lambda_1-\lambda_t)^2+\bar{a}_1^2)}{((\lambda_1-\lambda_t)^3+\bar{a}_1^2(\lambda_1-\lambda_t))^2}\lambda_t' > 0.$$

2. *Case 2: $\exists j \geq 2$ such that $\bar{a}_1\bar{a}_j \neq 0$.* We have, using once again equations (3.47)

and (3.48),

$$\begin{aligned}
\|x(t)\|^2 &= \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^2}, \\
\frac{d\|x(t)\|^2}{dt} &= 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^3}\lambda_t' = \frac{2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_t)^3}}{1+\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_t)^2}}, \\
\frac{d^2\|x(t)\|^2}{dt^2} &= \frac{\left(6\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_t)^4}\left(1+\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_t)^2}\right) - 4\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_t)^3}\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_t)^3}\right)\lambda_t'}{\left(1+\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_t)^2}\right)^2}.
\end{aligned}$$

From equations (3.53) and (3.54), our result is proved if we can show for all $\lambda \in (\lambda_1, \lambda_2)$, that

$$3\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^4}\left(1+\sum_{j=1}^{n} \frac{\bar{a}_j^2}{(\lambda_j - \lambda)^2}\right) - 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^3}\sum_{j=1}^{n} \frac{\bar{a}_j^2}{(\lambda_j - \lambda)^3}$$

is strictly positive. In fact, we prove the stronger statement, for $\lambda \in (\lambda_1, \lambda_2)$, that

$$\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^4}\sum_{j=1}^{n} \frac{\bar{a}_j^2}{(\lambda_j - \lambda)^2} - \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^3}\sum_{j=1}^{n} \frac{\bar{a}_j^2}{(\lambda_j - \lambda)^3} \tag{3.67}$$

is strictly positive. We may rewrite (3.67) as

$$\begin{aligned}
\sum_{i,j=1}^{n} \frac{\bar{a}_i^2\bar{a}_j^2}{(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(1 - \frac{\lambda_i - \lambda}{\lambda_j - \lambda}\right) &= \sum_{i,j=1}^{n} \frac{\bar{a}_i^2\bar{a}_j^2}{(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda}\right) \\
&= \sum_{i,j=1, i\neq j}^{n} \frac{\bar{a}_i^2\bar{a}_j^2}{(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda}\right).
\end{aligned}$$

The previous sum may be rewritten as

$$\begin{aligned}
&\sum_{j=2}^{n}\left\{\frac{\bar{a}_1^2\bar{a}_j^2}{(\lambda_1 - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_j - \lambda_1}{\lambda_j - \lambda}\right) + \frac{\bar{a}_j^2\bar{a}_1^2}{(\lambda_j - \lambda)^4(\lambda_1 - \lambda)^2}\left(\frac{\lambda_1 - \lambda_j}{\lambda_1 - \lambda}\right)\right\} + \\
&\sum_{i=2}^{n-1}\sum_{j>i}\left\{\frac{\bar{a}_i^2\bar{a}_j^2}{(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda}\right) + \frac{\bar{a}_j^2\bar{a}_i^2}{(\lambda_j - \lambda)^4(\lambda_i - \lambda)^2}\left(\frac{\lambda_i - \lambda_j}{\lambda_i - \lambda}\right)\right\}.
\end{aligned}$$

Recall, from Assumption 3.2.2, $\lambda_1 < \lambda_2$. Thus, the first sum is strictly positive for $\lambda \in (\lambda_1, \lambda_2)$, where we use that fact there exists $j \geq 2$ such that $\bar{a}_1 \bar{a}_j \neq 0$. We next claim, for $2 \leq i \leq n-1$ and $i < j \leq n$, that

$$\frac{\bar{a}_i^2 \bar{a}_j^2}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^2} \left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda} \right) + \frac{\bar{a}_j^2 \bar{a}_i^2}{(\lambda_j - \lambda)^4 (\lambda_i - \lambda)^2} \left(\frac{\lambda_i - \lambda_j}{\lambda_i - \lambda} \right) \tag{3.68}$$

is positive. Indeed, if $\bar{a}_i \bar{a}_j = 0$ or $\lambda_i = \lambda_j$, it is trivial. Otherwise, $\bar{a}_i \bar{a}_j \neq 0$ and $\lambda_i < \lambda_j$ and (3.68) is positive if and only if

$$\frac{1}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^3} - \frac{1}{(\lambda_j - \lambda)^4 (\lambda_i - \lambda)^3}$$

is positive. Rewriting the last expression, we obtain

$$\frac{\lambda_j - \lambda_i}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^4},$$

which is positive. Thus (3.67) is strictly positive and $\|x(t)\|^2$ is a strictly convex function for $t \in \mathcal{T}$.

□

Just as in Lemma 3.6, our next lemma shows $\|x(t)\|$ is related to the first derivative of m.

Lemma 3.14. *Let $t \in \mathcal{T}$. Then*

$$\|x(t)\| > (=, <)\ 1 \iff m'(t) < (=, >)\ 0. \tag{3.69}$$

Proof. For $t \in \mathcal{T}$, we have

$$\|x(t)\|^2 = \frac{z^2}{y_0^2} = \frac{1 - y_0^2}{y_0^2} = \frac{1 - \frac{m'(t)+1}{2}}{\frac{m'(t)+1}{2}} = \frac{1 - m'(t)}{1 + m'(t)}, \tag{3.70}$$

where the second equality follows from y being unit norm and where the third equality follows from equation (3.56). This proves our result since, for $t \in \mathcal{T}$, $m'(t) \in (-1, 1]$ and since the function $w(x) := \frac{1-x}{1+x}$ is strictly decreasing with $w(0) = 1$. □

Following are two theorems, analogous to Theorems 3.4 and 3.5, which relate a local non-global minimizer to the function m. The first one states that if a local-nonglobal minimizer of problem (2.1) or (2.2) exists, then there exists t^* where the first and second order optimality conditions for a local minimizer of m are satisfied. The second one is almost its converse: if the first and second order sufficient optimality conditions for a local minimizer of m are satisfied at some t^*, then $x(t^*)$ is the local-nonglobal minimizer of problem (2.2). It is also the local-nonglobal minimizer of problem (2.1), if the sign of the Lagrange multiplier is strictly negative.

Theorem 3.8. *Suppose x^* is a local-nonglobal minimizer of* (2.2) *or* (2.1) *with corresponding Lagrange multiplier λ^*. Let t^* be the solution to $\lambda_2(D(t)) = \lambda^*$. Then $m'(t^*) = 0$ and $m''(t^*) \geq 0$.*

Proof. By Lemma 3.1, $\lambda_2(D(t^*)) \in (\lambda_1(A), \lambda_2(A))$ and thus $t^* \in \mathcal{T}$. Theorem 3.7 gives $x^* = x(t^*)$. The fact that $m'(t^*) = 0$ follows from the feasibility of x^* and from Lemma 3.14.

By equations (3.54) and (3.63), and since, by Theorem 3.2, $\varphi'(\lambda^*) \leq 0$, we obtain

$$\frac{d\|x(t^*)\|^2}{dt} = \frac{d\varphi(\lambda_2(D(t^*)))}{d\lambda} \frac{d\lambda_2(D(t^*))}{dt} \leq 0. \tag{3.71}$$

If $m''(t^*) < 0$, then $m'(t^* - h) > 0$ for $h > 0$ small enough and using Lemma 3.14 we deduce $\|x(t^* - h)\| < 1$. Thus, since $\|x(t^*)\| = 1$,

$$\frac{d\|x(t^*)\|^2}{dt} = \lim_{h \to 0} \frac{\|x(t^*)\|^2 - \|x(t^* - h)\|^2}{h} \geq 0. \tag{3.72}$$

Inequalities (3.71) and (3.72) give

$$\frac{d\|x(t^*)\|^2}{dt} = 0.$$

It follows then from equality (3.71), and $\lambda_2'(D(t^*)) < 0$ that $\varphi'(\lambda_2(D(t^*))) = 0$. From equation (3.6), φ is strictly convex over the interval $(\lambda_1(A), \lambda_2(A))$ and thus $\lambda_2(D(t^*))$

is its strict minimizer. By equation (3.62), the following inequality thus holds

$$\|x(t)\| \geq \|x(t^*)\| = 1 \text{ for } t \in \mathcal{T}.$$

This contradicts $\|x(t^* - h)\| < 1$ for $h > 0$ small enough. Thus $m''(t^*) \geq 0$. □

Theorem 3.9. *Suppose $t^* \in \mathbb{R}$ satisfies $m'(t^*) = 0$ and $m''(t^*) > 0$, then $x(t^*)$ is a strict local non-global minimizer of (2.2) with Lagrange multiplier $\lambda^* := \lambda_2(D(t^*))$. In addition, if $\lambda^* < 0$, then $x(t^*)$ is a strict local-nonglobal minimizer of (2.1).*

Proof. From Lemma 3.12, $m''(t^*) \neq 0$ implies $\lambda_2(D(t^*)) \in (\lambda_1(A), \lambda_2(A))$, i.e. $t^* \in \mathcal{T}$. Therefore $x(t^*)$ is well defined and if we let $x^* := x(t^*)$ and $\lambda^* := \lambda_2(D(t^*))$, then by equation (3.61), the stationarity condition (3.1) is satisfied. Feasibility of x^* follows from Lemma 3.14. If we can further show $\varphi'(\lambda^*) < 0$, then the result follows from item 3 of Theorem 3.2.

Since $m''(t^*) > 0$, then $m'(t^* + h) > m'(t^*) = 0$ for $h > 0$ small enough. By Lemma 3.14, this implies $\|x(t^* + h)\| < 1$ and thus

$$\frac{d\|x(t^*)\|^2}{dt} = \lim_{h \to 0} \frac{\|x(t^* + h)\|^2 - \|x(t^*)\|^2}{h} \leq 0. \tag{3.73}$$

By a similar argument that appears in the proof of Theorem 3.8, we conclude that the inequality in (3.73) holds strictly. From inequality (3.54) and equation (3.63), we deduce $\varphi'(\lambda^*) < 0$. □

Bounds on λ^* and t^*

For this section, we assume a local-nonglobal minimizer of problem (2.2) exists. Our algorithm is based on finding a root t^* to $\|x(t)\|^2 - 1$ and we need initial bounds on t^*.

Lemma 3.15. *If a local-nonglobal minimizer of problems (2.1) or (2.2) exists, let t^* be defined as in Theorem 3.7. Then*

$$t^* \in (-\|a\| + \lambda_1(A), \|a\| + \lambda_2(A)).$$

Proof. By definition of t^*, $\lambda_2(D(t^*)) \in (\lambda_1(A), \lambda_2(A))$ and thus $t^* \in \mathcal{T}$. Hence, for $t = t^*$, if y satisfies equation (3.55), then equation (3.60) implies $y_0 \neq 0$. Dividing equation (3.57) by y_0, we obtain

$$t^* = a^T x^* + \lambda_2(D(t^*)) = a^T x^* + \lambda^*.$$

The conclusion follows from $\lambda^* \in (\lambda_1(A), \lambda_2(A))$ and $-\|a\| \leq a^T x^* \leq \|a\|$. □

Corollary 3.6. *If a local-nonglobal minimizer of problems* (2.1) *or* (2.2) *exists, then*

$$\lambda^* \in [\lambda_2(D(-\|a\| + \lambda_1(A))), \lambda_2(D(\|a\| + \lambda_2(A)))]\,. \tag{3.74}$$

Proof. Recall, from Lemma 3.12, $\lambda_2(D(t))$ is an increasing function. □

The Algorithm

We now describe our second algorithm for either computing a possible local-nonglobal minimizer of problem (2.2) or either returning as an output that such a candidate does not exist. We mention below how to modify the algorithm to compute local-nonglobal minimizer of problem (2.1). Since $\|x(t)\|^2$ is strictly convex for $t \in \mathcal{T}$ and since we have a lower bound on t^* when a local-nonglobal minimizer exists, it is not surprising this algorithm is similar in structure to Algorithm 3.2.1. To simplify our analysis, let $r(t) := \|x(t)\|^2 - 1$ and recall we are looking for a root of this function.

Algorithm 3.2.2.

1. INITIALIZATION
   - 1.1. Let $t_L = -\|a\| + \lambda_1(A)$, $t_U = \|a\| + \lambda_2(A)$, $t_0 = t_L - 0.1$, $t_1 = t_L$, $k = 1$.
   - 1.2. If $\lambda_2(D(t_L)) = \lambda_2(A)$ or if $\frac{r(t_1)-r(t_0)}{t_1-t_0} \geq 0$, LNGM = 0, else LNGM = 1.
2. ITERATION While $LNGM = 1$ and $\|x(t_k)\| \neq 1$, do
   - 2.1. $t_{k+1} = t_k - \frac{r(t_k)(t_k-t_{k-1})}{r(t_k)-r(t_{k-1})}$.
   - 2.2. If either $\lambda_2(D(t_{k+1})) = \lambda_2(A)$, $\frac{r(t_{k+1})-r(t_k)}{t_{k+1}-t_k} \geq 0$ or $t_{k+1} > t_U$, then

LNGM = 0.

2.3. $k = k + 1$.

The convergence results of Algorithm 3.2.2 and their proofs are identical to those of Theorem 3.6 and Corollary 3.4. We again suppose Assumption 3.2.3 holds.

Theorem 3.10. *The sequence $\{t_k\}$ produced by Algorithm 3.2.1 either converges to t^* such that $x(t^*)$ is a local-nonglobal minimizer of problem* (2.2) *or there does not exist a local-nonglobal minimizer of problem* (2.2) *and LNGM is set to 0.*

Corollary 3.7. *Suppose x^* is a local-nonglobal minimizer of problem* (2.2) *with a corresponding Lagrange multiplier λ^* that satisfies* (3.1) *. Let t^* be the unique solution to $\lambda_2(D(t)) = \lambda^*$. Then if $r'(t^*) > 0$, the sequence $\{t_k\}$ produced by Algorithm 3.2.2 converges to t^* superlinearly and $x(t^*)$ is a strict local-nonglobal minimizer of problem* (2.2).

Similarly to the comments that follow Corollary 3.4, we can modify Algorithm 3.2.2 in order to compute a local-nonglobal minimizer of problem (2.1) by setting the parameter LNGM to 0 if at some iteration k

$$\lambda_2(D(t_k)) \geq 0. \tag{3.75}$$

Again, with this change to the algorithm, Theorem 3.10 and Corollary 3.7 also hold if problem (2.2) is replaced by problem (2.1) in the statements.

3.3 Numerical Results

In this section, we compare Algorithms 3.2.1, 3.2.2 and the Martínez Algorithm [29]. As in Section §2.10, all algorithms were implemented using MATLAB 6.5 and computations were done on a Pentium 4 at 1.8GHz with 256MB of memory (all codes

may be found at the following URL: www.math.mcgill.ca/~fortin). The necessary eigenvalues and eigenvectors for Algorithm 3.2.1 are obtained, with the function `eigs`, using the matrix-vector multiplications stated in Section §2.3. For Algorithm 3.2.2, we naturally use the matrix-vector multiplications $w \rightarrow D(t)u$. Thus these algorithms are able to exploit the sparsity of the matrix A for large-sparse problems. On the other hand, the Martínez Algorithm requires at each iteration a matrix factorization (LDL^T or LU) or a spectral decomposition of a parameterized matrix of the same dimension as A. Therefore, this algorithm does not take advantage of sparsity. We should thus expect a great deal of improvement in the computation times of the matrix-free Algorithms 3.2.1 and 3.2.2.

For problems (2.2) of dimension n = 40, 80, 160, 320, 640 and 1280, we compared all three algorithms. For each dimension, five random problems were solved and data collected were averaged out. We also considered large problems of dimension n = 2000, 8000, 32 000 and 128 000, but only compared in these cases the matrix-free Algorithms 3.2.1 and 3.2.2. All problems considered had a density of $5/n$ and were generated so that a local-nonglobal minimizer existed (see Appendix A for the MATLAB source code used to generate the problems).

We coded the Martínez Algorithm (computing an LU factorization at each iteration and using the function S_1 for the parameter update, see [29]). All three algorithms halt when an approximate multiplier $\tilde{\lambda} \in (\lambda_1(A), \lambda_2(A)$ satisfy approximately

$$\|(A - \tilde{\lambda}I)^{-1}a\| \approx 1.$$

In our tests, the stopping criterion is

$$|\|(A - \tilde{\lambda}I)^{-1}a\| - 1| \leq 10^{-12}.$$

Results are shown on Figure 3.4 and Table 3.1, where we are interested in the number of iterations taken to converge, the total computation time and the number of matrix-

vector multiplications needed by the matrix-free algorithms to halt.

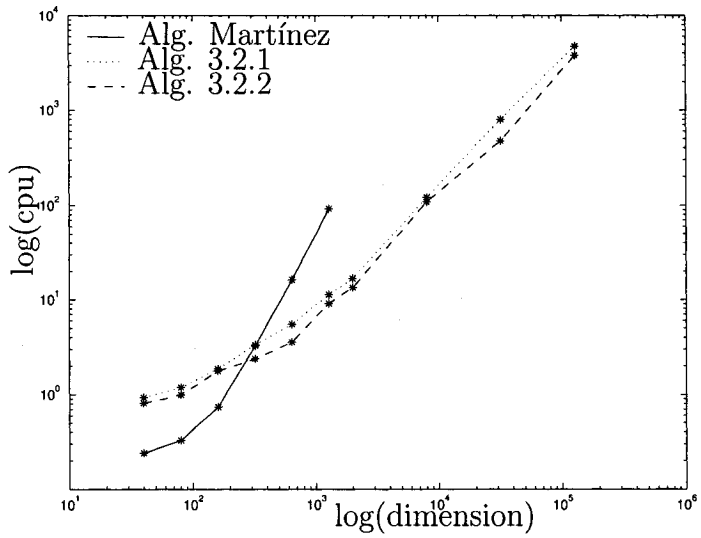

Figure 3.4: Logarithm of computation time required by the Martínez Algorithm, Algorithms 3.2.1 and 3.2.2 in function of the logarithm of problem dimensions.

We observe from Figure 3.4 that computation times are faster for small size problems for the Martínez Algorithm, even though the number of iterations taken to converge is approximately the double of what is required by the other algorithms. Martínez [29] claims local cubic convergence for his algorithm, but it may take a few iterations for convergence to take place. As for our algorithms, the approximate solution is better at each iteration and (superlinear) convergence is quickly attained.

However, for medium size sparse problems the Martínez Algorithm is outperformed by the matrix-free Algorithms 3.2.1 and 3.2.2. For all algorithms, the results indicate the computation time are proportional to a power of the problem dimensions.

When comparing Algorithms 3.2.1 and 3.2.2, we see both algorithms take approximately the same amount of iterations to converge, although these data are slightly better for Algorithm 3.2.1. However, Algorithm 3.2.2 clearly needs less matrix-vector multiplications at each step, a consequence of the simpler (symmetric) eigenvalue

	Iterations			Matrix-Vector ×	
Size(n)	Martínez	Alg. 3.2.1	Alg. 3.2.2	Alg. 3.2.1	Alg. 3.2.2
40	32.8	13.0	14.0	1180.4	988.2
80	33.8	13.0	14.0	1444.4	1176.0
160	33.8	14.2	15.2	2095.0	1975.8
320	37.2	15.4	16.4	3209.0	2242.2
640	40.0	16.0	17.0	3909.4	2557.6
1280	41.6	17.8	18.8	5320.6	4391.6
2000	-	15.8	16.8	5390.8	4571.4
8000	-	22.0	23.4	11061.2	10573.4
32000	-	21.2	22.0	12805.2	7983.2
128000	-	23.0	23.8	11577.6	9756.4

Table 3.1: Total number of iterations needed to converge by the Martínez Algorithm, Algorithms 3.2.1 and 3.2.2 in function of problem dimensions. The total number of matrix-vector multiplications is given for the last two matrix-free algorithms.

problems it needs to solve at each iterations). Therefore, even if the exponential growth in computation times is approximately of the same order for both algorithms, Algorithm 3.2.2 is faster.

Conclusion

We have considered in this thesis new approaches for computing the local minimizers of the trust-region subproblems (2.1) and (2.2). The algorithms presented have a different flavor depending on whether we seek for a global minimizer or a local non-global minimizer. In Algorithm 2.6.1 we presented a method for computing an approximate global minimizer which is a primal-dual method similar to the Rendl-Wolkowicz Algorithm and based implicitly on solving a semidefinite program. The algorithms we presented for computing an approximate local non-global minimizer are based upon finding a root of a convex function with the secant method. Our goal was to produce algorithms that only required matrix-vector multiplications (as opposed to matrix factorization or full spectral decomposition) and computing a few eigenvalues at each iteration, because we wanted to exploit the possible sparsity of the matrix A in the quadratic objective.

In Chapter 2 we have focused on finding an approximate solution to the equality constrained trust-region subproblem (2.2). By convexifying the quadratic objective using the feasibility constraint, we have formulated the problem of finding a global minimizer equivalently as the one of finding the intersection of the unit sphere and the largest volume ellipsoid inscribed in the unit ball among the family of ellipsoids (2.9). In our view, one of the interesting results of this chapter is that we can model this geometric formulation of the problem as a linear semidefinite program, different from the semidefinite programming relaxation usually associated with problem (2.2).

Although, our algorithm does not explicitly solve this SDP, it is implicitly solved by equivalently maximizing a concave function of a single variable over a closed interval. This is similar to what is done in the Rendl-Wolkowicz Algorithm. However, there is more link between the two algorithms, since there is a bijection between the results obtained in their framework and the ones obtained in ours. Therefore, both algorithms are in structure similar. This includes a similar way of detecting an interior optimal solution for problem (2.1) and a similar treatment of the hard case, i.e. when we are faced with non-differentiability of the single variable concave objective. In particular, we take advantage of the shift and deflate procedure introduced in [10]. The main drawback of our method is that we need to find at each iteration the smallest eigenvalue of a non symmetric parameterized matrix, where as, in the Rendl-Wolkowicz Algorithm, the parameterized matrix is symmetric. Thus there is at least more work involved in our method. In this respect, our algorithm is not computationally attractive. What we believe to be the main interest of this chapter lies in the fact that we have showed that two approaches, based on different semidefinite programs, lead to two algorithms that are quite similar in structure.

We introduced as well, in this chapter, Algorithm 2.6.2, a modified version of Algorithm 2.6.1, in order to prove convergence results that are useful when one is interested in proving the global convergence of a trust-region method for unconstrained optimization. Such problems are solved (we also solve constrained optimization problems) in the last part of our numerical results, but we were first interested in comparing Algorithm 2.6.1 with other existing approaches. Overall, our observations lead to the conclusion that this algorithm and the Rendl-Wolkowicz Algorithm behave similarly, although the latter algorithm is usually faster due to less work needed in solving the eigenvalue problems. We have also shown in Sections §2.10.1 and 2.10.1 examples where these algorithms outperform the approach of Moré-Sorensen which computes a Cholesky factorization at each iteration. However, Section 2.10.1 presents as well

cases where this is not always true and where sparsity is preserved through the factorizations. Furthermore, the results indicate that there is a trade-off between speed and robustness: on some problems the GLTR or DCA Algorithms perform impressively faster and necessitate less matrix-vector multiplication, but are either unable to obtain accurate solutions to hard case 2ii) TRS or convergence is extremely slow. However, the robustness of Algorithm 2.6.1 and the Rendl-Wolkowicz Algorithm is quite dependent on the eigensolver used. We must admit to be disappointed by the fact that `eigs` sometimes failed to return accurate eigenvalues and eigenvectors or to converge and that we had to rely on `eig`. This had the effect of limiting the size of our tests problems.

In Chapter 3, we have considered two algorithms for computing the local-nonglobal minimizer of problem (2.1) or (2.2). For each algorithm, the main computing effort at each step lies in approximating the first two eigenvalues of the same parameterized matrices found in Algorithm 2.6.1 and the Rendl-Wolkowicz Algorithm. By extending the geometric approach of Chapter 2, we have showed a relationship between global minimizers and the local-nonglobal minimizer: each minimizer lies in the intersection of the unit sphere with an ellipsoid locally contained in the unit ball at the minimizer. As we have seen in Section §3.2.1, this is a generalization of the result of Martinez [29] that was known to hold when the trust-region radius tends to infinity.

Algorithms 3.2.1 and 3.2.2 presented in this chapter are respectively motivated by Corollary 2.1 and Cauchy's inequality, Lemma 3.11, and build on the theory needed to derive Algorithm 2.6.1 and the Rendl-Wolkowicz Algorithm. Both algorithms are based on applying the secant method respectively to the strictly convex functions $\|x(\gamma)\|^2 - 1$ and $\|x(t)\|^2 - 1$. Although the author was not able to prove it, it seems the functions $\|x(\gamma)\| - 1$ and $\|x(t)\| - 1$ are strictly convex as well, and numerical experiments tend to show the performance of the secant method on these functions is enhanced. Obviously this constitute material for future research. In any case, our

numerical results clearly show our approaches are more competitive than the previous approach of Martinez. Because the eigenvalue problems that need to be solved in Algorithm 3.2.2 are simpler, less matrix-vector multiplications and computation time are required compared to Algorithm 3.2.1.

For final remarks, we would like to point out some additional work that could be done in order to improve the results of this thesis. Algorithm 2.6.1 is based on finding at each iteration a nearly exact smallest eigenvalue. Is it possible to derive an algorithm that only requires approximate eigenvalues? At the moment, finding nearly exact eigenvalues seem to require on some problems many matrix-vector multiplications and thus affect the computation times. We may also see this problem the other way around and wonder if it is possible to set the eigenvalue problems in a way that facilitates computations by ARPACK (`eigs`)? Obviously, similar remarks apply to the algorithms of Chapter 3.

Bibliography

[1] K. ANSTREICHER and H. WOLKOWICZ. On Lagrangian relaxation of quadratic matrix constraints. *SIAM J. Matrix Anal. Appl.*, 22(1):41–55 (electronic), 2000.

[2] A. BEN-TAL and A. NEMIROVSKI. *Lectures on modern convex optimization.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. Analysis, algorithms, and engineering applications.

[3] R. BHATIA. *Matrix analysis*, volume 169 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1997.

[4] S. BOYD, L. EL GHAOUI, E. FERON, and V. BALAKRISHNAN. *Linear matrix inequalities in system and control theory*, volume 15 of *SIAM Studies in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[5] A. L. CAUCHY. Sur l'équation à l'aide de laquelle on détermine les inégalités séculaires des mouvements des planètes. In *Oeuvres Complètes (II^e Série)*, volume 9. 1829.

[6] M. CELIS, J. DENNIS, and R. TAPIA. A trust region strategy for nonlinear equality constrained optimization. In *Numerical optimization, 1984 (Boulder, Colo., 1984)*, pages 71–82. SIAM, Philadelphia, PA, 1985.

[7] A. CONN, N. GOULD, D. ORBAN, and P. TOINT. A primal-dual trust-region algorithm for non-convex nonlinear programming. *Math. Program.*, 87(2, Ser. B):215–249, 2000. Studies in algorithmic optimization.

[8] A. CONN, N. GOULD, and P. TOINT. *Trust-region methods.* MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[9] C. FORTIN. A survey of the trust region subproblem within a semidefinite framework. Master's thesis, University of Waterloo, 2000.

[10] C. FORTIN and H. WOLKOWICZ. The trust region subproblem and semidefinite programming. *Optimization Methods and Software*, 19(1):41–67, February 2004.

[11] W. GANDER. Least squares with a quadratic constraint. *Numer. Math.*, 36(3):291–307, 1980/81.

[12] D. GAY. Computing optimal locally constrained steps. *SIAM J. Sci. Statist. Comput.*, 2:186–197, 1981.

[13] D. GAY. Computing optimal locally constrained steps. *SIAM Journal on Scientific and Statistical Computing*, 2(2):186–197, 1981.

[14] S. GOLDFELD, R. QUANDT, and H. TROTTER. Maximization by quadratic hill-climbing. *Econometrica*, 34(2):541–551, 1966.

[15] G. GOLUB and C. VAN LOAN. *Matrix Computation.* The Johns Hopkins University Press, third edition, 1996.

[16] G. GOLUB and U. VON MATT. Quadratically constrained least squares and quadratic problems. *Numer. Math.*, 59(6):561–580, 1991.

[17] N. GOULD, S. LUCIDI, M. ROMA, and P. TOINT. Solving the trust-region subproblem using the lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

[18] N. GOULD, D. ORBAN, and P. TOINT. Cuter: Constrained and unconstrained testing environment revisited. http://cuter.rl.ac.uk/cuter-www/index.html, 2001.

[19] W. HAGER. Minimizing a quadratic over a sphere. *SIAM J. Optim.*, 12(1):188–208 (electronic), 2001.

[20] M. HEINKENSCHLOSS. Mesh independence for nonlinear least squares problems with norm constraints. *SIAM J. Optim.*, 3(1):81–117, 1993.

[21] M. HEINKENSCHLOSS. On the solution of a two ball trust region subproblem. *Math. Programming*, 64(3, Ser. A):249–276, 1994.

[22] A. HOERL. and R. KENNARD. Ridge regression: Biased estimation of nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[23] R. HORN and C. JOHNSON. *Matrix Analysis.* Cambridge University Press, 1987.

[24] R. LEHOUCQ and D. SORENSEN. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl.*, 17(4):789–821, 1996.

[25] R. LEHOUCQ, D. SORENSEN, and C. YANG. *ARPACK users' guide.* Software, Environments, and Tools. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.

[26] K. LEVENBERG. A method for the solution of certain nonlinear problems. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[27] S. LUCIDI, L. PALAGI, and M. ROMA. On some properties of quadratic programs with a convex quadratic constraint. *SIAM J. Optim.*, 8(1):105–122 (electronic), 1998.

[28] D. MARQUARDT. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[29] J. MARTÍNEZ. Local minimizers of quadratic functions on euclidian balls and spheres. *SIAM Journal on Optimization*, 4(1):159–176, 1994.

[30] J. MARTÍNEZ and S. SANTOS. A trust-region strategy for minimization on arbitrary domains. *Math. Programming*, 68(3, Ser. A):267–301, 1995.

[31] J. MORÉ. Generalization of the trust region problem. *Optimization Methods and Software*, 2:189–209, 1993.

[32] J. MORÉ and D. SORENSEN. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[33] A. NEUMAIER. *Introduction to numerical analysis.* Cambridge University Press, Cambridge, 2001.

[34] J. NOCEDAL and S.J.WRIGHT. *Numerical optimization.* Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[35] J.-M. PENG and Y.-X. YUAN. Optimality conditions for the minimization of a quadratic with two quadratic constraints. *SIAM J. Optim.*, 7(3):579–594, 1997.

[36] M. POWELL. A new algorithm for unconstrained optimization. In J. Rosen, O. Mangasarian, and K. Ritter, editors, *Nonlinear Programming*, pages 31–65. Academic Press, New York, NY, 1970.

[37] M. POWELL. Convergence properties of a class of minimization algorithms. In O. Mangasarian, R. Meyer, and S. Robinson, editors, *Nonlinear Programming 2*, pages 1–27. Academic Press, New York, NY, 1975.

[38] M. POWELL and Y. YUAN. A trust region algorithm for equality constrained optimization. *Math. Programming*, 49(2, (Ser. A)):189–211, 1990/91.

[39] F. RENDL and H. WOLKOWICZ. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming Series B*, 77(2):273–299, 1997.

[40] M. ROJAS, S. SANTOS, and D. SORENSEN. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM J. Optim.*, 11(3):611–646 (electronic), 2000/01.

[41] M. ROJAS and D. SORENSEN. A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems. *SIAM J. Sci. Comput.*, 23(6):1842–1860 (electronic), 2002.

[42] D. SORENSEN. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982.

[43] D. SORENSEN. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13(1):357–385, 1992.

[44] T. STEIHAUG. The conjuguate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3), 1983.

[45] R. STERN and H. WOLKOWICZ. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optim.*, 5(2):286–313, 1995.

[46] J. STURM and S. ZHANG. On cones of nonnegative quadratic functions. *Math. Oper. Res.*, 28(2):246–267, 2003.

[47] P. TAO and L. AN. Difference of convex functions optimization algorithms (dca) for globally minimizing nonconvex quadratic forms on euclidean balls and spheres. *Oper. Res. Lett*, 19(5):207–216, 1996.

[48] P. TAO and L. AN. A d.c. optimization algorithm for solving the trust-region subproblem. *SIAM J. Optim.*, 8(2):476–505 (electronic), 1998.

[49] P. TOINT. Towards an efficient sparsity exploiting newton method for minimization. In I. S. Duff, editor, *Sparse matrices and their uses*, Institute of Mathematics and its Applications Conference Series, pages xii+387, London, 1981. Academic Press Inc. [Harcourt Brace Jovanovich Publishers].

[50] H. WEYL. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen. *Mathematische Annalen*, 71:441–469, 1911.

[51] J. WILKINSON. *The algebraic eigenvalue problem.* Clarendon Press, Oxford, 1965.

[52] H. WOLKOWICZ. Measures for symmetric rank-one updates. *Math. Oper. Res.*, 19(4):815–830, 1994.

[53] Y. YE and S. ZHANG. New results on quadratic minimization. *SIAM J. Optim.*, 14(1):245–267 (electronic), 2003.

[54] Y. YUAN. On a subproblem of trust region algorithms for constrained optimization. *Math. Programming*, 47(1, (Ser. A)):53–63, 1990.

[55] Y. YUAN. A review of trust region algorithms for optimization. In *ICIAM 99 (Edinburgh)*, pages 271–282. Oxford Univ. Press, Oxford, 2000.

Appendix A

Matlab functions

A function used for creating hard case 2ii) TRS with $\log(n)$ dense rows and columns.

```
function [A,a,s]=genhard(n,density)
% generates a random hard case 2ii) TRS.
nrow = floor(log(n));
tt=1/rand;
D = tt*sprand(n-nrow,n-nrow,density);
D = D+D';
% fill the first 'nrows' rows and columns: to create problems for Cholesky factorization
E = sprandsym(nrow,1);
WW = rand(nrow,n-nrow);
A = [E,WW;WW',D];
tt=1/rand;
a=tt*sprandn(n,1,.5);
[ va,lambdaA ] = eigs(A,1,'SA');
if lambdaA ≥ 0
  shift = 10*1/rand;
  A = A-(lambdaA+shift)*speye(n);
```

```
 lambdaA = -shift;
end
atemp = a;
a=(A-lambdaA*speye(n))*a; % ensure hard case, i.e. a is in range
s=2*norm(atemp); % to ensure hard case 2ii), i.e. s too large
```

A function used for generating TRS with a local-nonglobal minimizer

```
function [A,a,s]=genlngm(n,density)
% generates a TRS with a local-nonglobal minimizer.
tt=1/rand;
A = tt*sprand(n,n,density);
A = A+A';
tt=1/rand;
a=tt*sprandn(n,1,density);
s=1;
% get the two smallest eigenvalues of A
OPTIONS.tol= eps;
OPTIONS.issym = 1; % because we know the matrices A and D are symmetric
OPTIONS.disp = 0; % no display of the output for eigs.m
[ v,lambda,flageigs ] = eigs(A,2,'SA',OPTIONS); % computes 2 smallest eigenvalues of A
lambda = diag(lambda);
% avoid the hard case
a = 1/n*v(:,1) + a; % enforces a'v(:,1)  = 0
% ensure a local nonglobal min exists.
na = norm(a);
while na > abs(lambda(2)-lambda(1))/2
```

Bibliography

[1] K. ANSTREICHER and H. WOLKOWICZ. On Lagrangian relaxation of quadratic matrix constraints. *SIAM J. Matrix Anal. Appl.*, 22(1):41–55 (electronic), 2000.

[2] A. BEN-TAL and A. NEMIROVSKI. *Lectures on modern convex optimization.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. Analysis, algorithms, and engineering applications.

[3] R. BHATIA. *Matrix analysis*, volume 169 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1997.

[4] S. BOYD, L. EL GHAOUI, E. FERON, and V. BALAKRISHNAN. *Linear matrix inequalities in system and control theory*, volume 15 of *SIAM Studies in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[5] A. L. CAUCHY. Sur l'équation à l'aide de laquelle on détermine les inégalités séculaires des mouvements des planètes. In *Oeuvres Complètes (II^e Série)*, volume 9. 1829.

[6] M. CELIS, J. DENNIS, and R. TAPIA. A trust region strategy for nonlinear equality constrained optimization. In *Numerical optimization, 1984 (Boulder, Colo., 1984)*, pages 71–82. SIAM, Philadelphia, PA, 1985.