Semidefinite Programming Applied to Nonlinear Programming

by

Serge G. Kruk

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Mathematics

in

Combinatorics and Optimization

Waterloo, Ontario, Canada, 1997

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Arguably the most successful family of techniques used to solve general nonlinear programs is known as Sequential Quadratic Programming. This iterative approach rests on a quadratic model of the Lagrangean subjected to linear approximations of the constraints. For all its success, the practical implementations must somehow overcome the weaker model of the feasible region.

A model demonstrably closer to the original problem uses second-order Taylor expansions of both objective function and constraints. Such a model preserves all curvature information and can therefore provide better Lagrange multipliers estimates. While considered before, this approach has generally been discarded as intractable. But the expanding field of semidefinite programming offers tools, both theoretical and practical, to overcome for a large class of problems this presumed intractability.

To introduce such tools in a setting other than the combinatorial optimization environment, where they have made notable breakthroughs in recent years, we review recent results concerning the Trust-Region Subproblem, a basic building block in the continuous optimization arena. The relation between Lagrangean and semidefinite duality is explored and leads to simple theoretical foundations of an easily implemented algorithm.

The trust-region subproblem is then generalized to a problem with multiple trust regions. In this case the feasible set is lifted from its the original Euclidean space to a symmetric matrix space, partially ordered by the semidefinite cone. This generalization leads ultimately to an algorithm engineered around the fully quadratic subproblem, envisioned as the better model within a sequential programming framework.

Contents

List of Figures

Chapter 1

The trust-region subproblem

As introduction to our work towards a general nonlinear program solver, we consider the Trust-Region Subproblem, hereafter *TRS* , described by

$$TRS \qquad \min\left\{\mu(x) = x^tQx + 2b^tx \;\middle|\; x^tx \leq \delta^2, x \in \mathbb{R}^n\right\},$$

where $\mathbb{R}^n$ is the n-dimensional Euclidean space and Q is in $\mathbb{S}_n$ the space of symmetric $n \times n$ matrices. This problem, as well as occurring in its own right under the names *ridge analysis* or *ridge regression* (see Hoerl [31], Draper [17]), arises in an important class of minimization algorithms known as trust-region or restricted-step methods described in most textbooks. (See, for example, Fletcher [20], Luenberger [37], Dennis and Schnabel [14] or Bertsekas [6].)

Originally considered by Levenberg [34] in 1944 and Marquadt [41] in the context of nonlinear least squares and expanded upon by Forsythe and Golub [22] in 1965 who characterized all stationary solutions, *TRS* has now been extensively studied and a number of algorithms can solve very efficiently most instances of the problem. (See, for example, Moré and Sorensen [44], Rendl and Wolkowicz [57], Santos and Sorensen [59], and Tao and An [67].) Our intention here is not to surpass these special-purpose algorithms but rather to show how semidefinite programming can be used as a framework both to study nonlinear programs and to produce very simple algorithms solving *TRS*. We then generalize this work; first to multiple trust-regions and then to general nonlinear programs.

The program *TRS*, as defined above, is a simplified form of a problem that appears in the modeling literature as

$$\min\left\{\mu(x) = x^tQx + 2b^tx \mid x^tCx + 2d^tx \leq \delta^2, x \in \mathbb{R}^n\right\},$$

where C is positive definite. The notation used throughout this work, $C \succ 0$ ($C \succeq 0$) refers to the Löwner partial order and is used to indicate that C is a positive definite (positive semidefinite) matrix. The problem appears in this form when two competing objectives must be managed concurrently. If, for example, a manufacturing process has a mean and a variance approximated by second-order polynomial regressions, and the goal is to minimize the mean while maintaining the variance to a particular target value, the above model ensues. (See, for example, Vinning and Myers [71].)

This is indeed equivalent to *TRS* since, for $C \succ 0$, we can obtain the Cholesky factorization $C = LL^t$ and the substitution $x = L^{-t}z + C^{-1}d$ will yield

$$\min\left\{\mu(z) = z^t\tilde{Q}z + 2\tilde{b}^tz + a \mid z^tz \leq \delta^2 + d^tC^{-1}d\right\},$$

where $\tilde{Q} = L^{-1}QL^{-t}$, $\tilde{b} = L^{-1}QC^{-1}d + L^{-1}b$ and where the last term, $a = -2b^tC^{-1}d$, is a constant that can be safely neglected.

In this chapter, we review the theoretical aspects of *TRS* within a semidefinite framework and solve the program in a simple, straight-forward way that generalizes to multiple trust regions.

1.1 Characterization of optimality

The first surprising aspect of the trust-region subproblem is that the standard necessary conditions, that require the Hessian of the Lagrangean to be positive semidefinite only on a subspace, do not tell the complete story. The actual necessary conditions of *TRS* are stronger than expected.

As we will see, the standard sufficient conditions for global minima are also necessary. This is unusual since, in nonlinear programming, we are generally satisfied with local optimality results and concerned with global optimality only when we consider convex programs.

To introduce this aspect of *TRS*, we closely follow Moré and Sorensen [44] and explicitly state the global sufficient conditions as they apply to *TRS*.

Lemma 1.1.1 *Suppose that a scalar $\lambda \geq 0$ and a feasible vector $x_\lambda \in \mathbb{R}^n$, satisfy*

$$(Q+\lambda I)x_\lambda = -b \qquad \textit{(stationarity)},$$

$$\lambda(x_\lambda^t x_\lambda - \delta^2) = 0 \qquad \textit{(complementarity)},$$

$$(Q+\lambda I) \succeq 0 \qquad \textit{(strengthened second-order)}.$$

The vector x_λ is then optimal for TRS. Moreover, if $(Q+\lambda I) \succ 0$, then x_λ is the unique minimizer.

Proof: Since we have $Q+\lambda I \succeq 0$, we may consider the convex program $\min\{x^t(Q+\lambda I)x + 2b^t x\}$. Because x_λ satisfies $(Q+\lambda I)x_\lambda = -b$, the sufficient conditions for an unconstrained minimization are met. Therefore, for any $x \in \mathbb{R}^n$, we have

$$\begin{array}{rrcl} & x^t(Q+\lambda I)x + 2b^t x & \geq & x_\lambda^t(Q+\lambda I)x_\lambda + 2b^t x_\lambda \\ \Longleftrightarrow & \mu(x) + \lambda x^t x & \geq & \mu(x_\lambda) + \lambda x_\lambda^t x_\lambda \\ \Longleftrightarrow & \mu(x) & \geq & \mu(x_\lambda) + \lambda(x_\lambda^t x_\lambda - x^t x). \end{array}$$

First, if $\lambda = 0$, then we have $\mu(x) \geq \mu(x_\lambda)$ and we conclude that x_λ is optimal for *TRS*. On the other hand, if $\lambda > 0$, by complementarity, which is assumed to hold, $x_\lambda^t x_\lambda = \delta^2$ and again $\mu(x) \geq \mu(x_\lambda)$ for all $x^t x \leq \delta^2$. Therefore x_λ is optimal for *TRS*. The first claim is established. Now, if $(Q+\lambda I) \succ 0$, the previous derivation can be reproduced with strict inequality and uniqueness of x_λ follows. □

We should note that the optimality of x_λ for *TRS* implies that x_λ is a global minimum, a feature usually reserved for convex programs. Since we have imposed no conditions on the objective function, *TRS* is not, in general, a convex program. We therefore should not expect the sufficient conditions to be necessary. Yet, this rather surprising result was established first by Gay [24], concurrently by Sorensen [62], and then by numerous other researchers.

The surprise concerns, of course, the number of negative eigenvalues of the Hessian, $Q + \lambda I)$. Throughout this work, to denote the eigenvalues of an $n \times n$ matrix H, we use $\lambda(H) := \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, where we assume the ordering $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

To simplify the development, the following corollary to the Courant-Fisher min-max Theorem helps and is used repeatedly.

Corollary 1.1.2 *If an $n \times n$ symmetric matrix H satisfies $y^t H y \geq 0$ for all y in a subspace V of dimension $n - k$, then H has at most k negative eigenvalues.*

Proof: We assume the above convention for the ordering of the eigenvalues of H and we invoke the Courant-Fisher Theorem, one formulation of which (see Horn and Johnson [32],page 179) reads

$$\lambda_{k+1} = \max \Big\{ \min \left\{ \frac{y^t A y}{y^t y} \mid y \neq 0, y \perp w_1, w_2, \ldots, w_k \right\} \mid w_1, w_2, \ldots, w_k \in \mathbb{R}^n \Big\}.$$

Since V is $(n-k)$-dimensional, its orthogonal complement, $V^\perp$, is k-dimensional and has a basis $w_1, w_2, \ldots, w_k$. This choice of vectors in the Courant-Fisher min-max program forces the inner minimization to be nonnegative since, for all vectors y such that $y \perp w_i$, by hypothesis, $y^t H y \geq 0$. Therefore, $\lambda_{k+1}, \lambda_{k+2}, \ldots, \lambda_n \geq 0$ and H can have at most k negative eigenvalues. □

Extending Gay [24] and Sorensen [62], we now characterize the necessary conditions for both local and global minima. A proof for the global minima can be found in Fletcher [21].

Lemma 1.1.3 *If x is a local minimum of TRS, then there exists a multiplier $\lambda \geq 0$ satisfying stationarity, complementarity and for which $Q + \lambda I$ has at most one negative eigenvalue. If x is a global minimum, then $Q + \lambda I$ is positive semidefinite.*

Proof: Consider the Lagrangean $\mathcal{L}(x, \lambda) = x^t Q x + 2b^t x + \lambda(x^t x - \delta^2)$. We assume here, and in the following, that the interior of the trust-region is non-empty (Slater's condition). This is a very weak assumption, equivalent to $\delta > 0$. The standard first-order necessary conditions yield a vector x and a nonnegative multiplier λ satisfying complementarity and where $0 = \nabla\mathcal{L}(x, \lambda) = 2Qx + 2b + 2\lambda$ or $(Q + \lambda I)x = -b$. Therefore, the first three conditions of the lemma are satisfied.

Now, the standard second-order condition (see, for example, Fletcher [21]) yields $y^t(Q+\lambda I)y \geq 0$ on the $(n-1)$-dimensional plane $\{y : y \perp x\}$ tangent to the active constraint at the optimal point. We can assume that the constraint is active since, if it is not, then $Q \succeq 0$, $\lambda = 0$ and we have a global minimum by the unconstrained optimality conditions. In the active constraint case, therefore, the Hessian, by Corollary 1.1.2, has at most one negative eigenvalue corresponding to eigenvector x, the local minimum.

There remains only to show that the strengthened second-order condition applies to the global minimizer. To establish the claim, consider first the case $x^t x < \delta^2$. The tangent plane is then $\mathbb{R}^n$ and we are done. If $x^t x = \delta^2$, then we consider the equality-constrained trust-region program

$$\textit{EQ-TRS} \qquad \min\left\{\mu(y) = y^t Q y + 2b^t y \mid y^t y = \delta^2\right\}.$$

Clearly, x, optimal for *TRS*, is also optimal for the equality-constrained variation since the feasible set of the latter is a strict subset of the former. For all y, feasible for *EQ-TRS*, we therefore have $\mu(y) \geq \mu(x)$. This relation expands to

$$y^t Q y + 2b^t y \geq x^t Q x + 2b^t x.$$

By stationarity, which holds, $(Q + \lambda I)x = -b$ and we can simplify to

$$\begin{aligned} & y^t Q y - 2x^t(Q + \lambda I)y & \geq & \quad x^t Q x - 2x^t(Q + \lambda I)x \\ \iff \quad & (y - x)^t(Q + \lambda I)(y - x) & \geq & \quad \lambda(y^t y - x^t x). \end{aligned}$$

Now, since the constraint was $y^t y = \delta^2$, and $\delta^2 = x^t x$, we can further reduce to

$$(y - x)^t(Q + \lambda I)(y - x) \geq 0.$$

This is enough to conclude that $Q + \lambda I$ is positive semidefinite. To verify the claim, consider an arbitrary vector $z \in \mathbb{R}^n$. If z is orthogonal to x, then by the standard second-order condition, since x is optimal, we have $z^t(Q + \lambda I)z \geq 0$. If z is not orthogonal to x, it can be written from two feasible vectors y and x as $z = \alpha(y - x)$, for some scalar α. This follows from the unique constraint of *EQ-TRS*: The vector y was restricted in *size* to the radius of the ball, but not in *direction*. We now derive the required inequality,

$$\begin{aligned} z^t(Q + \lambda I)z &= \alpha(y - x)^t(Q + \lambda I)\alpha(y - x) \\ &= \alpha^2(y - x)^t(Q + \lambda I)(y - x) \\ &\geq 0, \end{aligned}$$

and we conclude that $Q + \lambda I$ is positive semidefinite. □

We therefore have, for *TRS*, a complete characterization of both local and global solutions. The question arises now of the existence of such local, non-global solutions of *TRS* and of their effect on any algorithm. The existence question was answered by Martinez [42] directly from the primal problem. We will review this result in the next chapter. At this point, we are more concerned with completing our investigation of *TRS*, for which we need the dual program. Incidentally, developing this dual will also explains why *TRS* behaves as a convex program.

1.2 Lagrangean relaxation

The complete characterization of optimality with identical necessary and sufficient conditions is somewhat surprising as *TRS* is not, in general, a convex program. The reason for this fortunate coincidence was uncovered by Stern and Wolkowicz [64]: The *TRS* **is** a hidden convex program to which, therefore, strong duality applies.

To simplify the following development and provide a form of *TRS* better suited to analysis, we consider an orthogonal rotation of the space of the problem. This, of course, is not done numerically while solving *TRS*. It is used only to explicate the behavior.

Since Q is symmetric, we can find a spectral decomposition $Q = V^tDV$, where V is orthogonal and D is a diagonal matrix of eigenvalues i.e., $D = Diag(\lambda(Q))$. We now define $z = V^t x$ and $c = V^t b$ to get TRS in the following form:

$$TRS \qquad \min\left\{\mu(x) = z^t Dz + 2c^t z \mid z^t z \leq \delta^2, z \in \mathbb{R}^n\right\}.$$

The Lagrangean of this formulation of TRS is

$$\mathcal{L}(z, \lambda) = z^t Dz + 2c^t z + \lambda(z^t z - \delta^2).$$

Now, by maximizing the dual functional

$$\nu(\lambda) \quad := \quad \min\left\{\mathcal{L}(z, \lambda) \mid z \in \mathbb{R}^n\right\},$$

we can derive the Lagrangean dual program ,

$$\begin{aligned}
\nu^* &= \max\left\{\nu(\lambda) \mid \lambda \geq 0\right\} \\
&= \max\left\{\min\left\{z^t Dz + 2c^t z + \lambda(z^t z - \delta^2) \mid z \in \mathbb{R}^n\right\} \mid \lambda \geq 0\right\} \\
&= \max\left\{\min\left\{z^t (D + \lambda I)z + 2c^t z - \lambda\delta^2 \mid z \in \mathbb{R}^n\right\} \mid \lambda \geq 0\right\}.
\end{aligned}$$

The inner minimization must be bounded, as TRS has an optimal solution. There is therefore a hidden semidefinite constraint, which we can make explicit, to get

$$\nu^* = \max\left\{\min\left\{z^t (D + \lambda I)z + 2c^t z - \lambda\delta^2 \mid z \in \mathbb{R}^n\right\} \mid D + \lambda I \succeq 0, \lambda \geq 0\right\}.$$

Since the inner unconstrained minimization must satisfy stationarity, $(D + \lambda I)z = -c$, we can write an optimal solution z in terms of λ as $z_\lambda = -(D + \lambda I)^- c$, where $(\cdot)^-$ is any generalized $\{2\}$-inverse.

We say that A^- is a $\{2\}$-inverse of A, in the manner of Ben-Israel and Greville [4], if it satisfies the second Penrose equation,

$$A^- A A^- = A^-.$$

It is interesting, and useful, that any $\{2\}$-inverse produces the same optimal value. To verify this claim, consider the objective function, evaluated at an optimal solution,

$$\begin{aligned}\nu(\lambda) &= z_\lambda^t(D+\lambda I)z_\lambda + 2c^t z_\lambda - \lambda\delta^2 \\ &= c^t(D+\lambda I)^-(D+\lambda I)(D+\lambda I)^- c - 2c^t(D+\lambda I)^- c - \lambda\delta^2 \\ &= -c^t(D+\lambda I)^- c - \lambda\delta^2.\end{aligned}$$

Now, as $c \in \mathcal{R}(D+\lambda I)$ (the stationarity equation is consistent), an orthogonal projection of c onto this range will not affect it, (i.e., $P_{\mathcal{R}(D+\lambda I)}c = c$) and

$$\nu(\lambda) = -c^t P_{\mathcal{R}((D+\lambda I)^t)}(D+\lambda I)^- P_{\mathcal{R}(D+\lambda I)}c - \lambda\delta^2,$$

which, by the $\{2\}$-inverse property, yields,

$$\nu(\lambda) = -c^t(D+\lambda I)^\dagger c - \lambda\delta^2,$$

where $(\cdot)^\dagger$ indicates the Moore-Penrose inverse. (See Ben-Israel and Greville [4], page 70.) Since the optimal objective value is therefore independent of the choice of inverse, we can write an explicit dual program,

$$\begin{aligned}\text{Dual } \mathit{TRS} \qquad \nu^* &= \max\left\{\nu(\lambda) = -c^t(D+\lambda I)^\dagger c - \lambda\delta^2 \;\middle|\; D+\lambda I \succeq 0, \lambda \geq 0\right\} \\ &= \max\left\{\nu(\lambda) = \sum_{i=1}^n \frac{-c_i^2}{\lambda_i+\lambda} - \lambda\delta^2 \;\middle|\; D+\lambda I \succeq 0, \lambda \geq 0\right\}.\end{aligned}$$

From this explicit dual, Stern and Wolkowicz [64] derive the following surprising result.

Lemma 1.2.1 *(Strong duality) The dual optimal value is attained and is equal to the primal optimal value.*

Proof: By weak Lagrangean duality,

$$\begin{array}{rcl} \mu^* & = & \min\left\{\mu(z) = z^t D z + 2c^t z \mid z^t z \leq \delta^2\right\} \\ & \geq & \nu^* \;\; = \max\left\{\nu(\lambda) = \sum_{i=1}^{n} \frac{-c_i^2}{\lambda_i + \lambda} - \lambda\delta^2 \mid D + \lambda I \succeq 0, \lambda \geq 0\right\}. \end{array}$$

From the above formulation of the dual, the behavior of the dual functional $\nu(\lambda)$ on the domain of interest, namely $\lambda \geq \max\{0, -\lambda_1(D)\}$, is apparent. We first show that the optimum is attained. Let I be the index set of eigenvalues equal to the smallest, i.e., $I = \{i : \lambda_i = \lambda_1(D)\}$. Consider $\nu(\lambda)$ from $-\lambda_1(D)$ to ∞, a larger set than the feasible region,

$$\begin{array}{rcl} \lim_{\lambda \to \infty} \nu(\lambda) & = & -\infty \\ \lim_{\lambda + \to -\lambda_1(D)} \nu(\lambda) & = & \begin{cases} -\infty, & \text{if } c_i \neq 0 \text{ for some } i \in I; \\ \text{finite}, & \text{if } c_i = 0 \text{ for all } i \in I. \end{cases} \end{array}$$

The dual objective then, if it has a discontinuity at $-\lambda_1$, is coercive and attains its maximum. If it has no discontinuity, the maximum is also attained, either at a stationary point $\lambda^* > -\lambda_1$ in the so-called easy hard-case, or at $\lambda^* = -\lambda_1$ in the hard sub-case of the hard-case.

We now differentiate ν to investigate the optimal objective value. The difficulty lies in differentiating the inverse. The following transformation is helpful. Recall that $(D + \lambda I)z_\lambda = -c$. We can differentiate with respect to λ to get $z_\lambda + (D + \lambda I)z'_\lambda = 0$ and then rewrite the dual objective as

$$\nu(\lambda) = z_\lambda^t (D + \lambda I)(D + \lambda I)^\dagger c - \lambda\delta^2 = z_\lambda^t c - \lambda\delta^2.$$

We can now obtain the first derivative,

$$\begin{array}{rcl} \nu'(\lambda) & = & c^t z'_\lambda - \delta^2 = -c^t (D + \lambda I)^\dagger z_\lambda - \delta^2 \\ & = & z_\lambda^t (D + \lambda I)(D + \lambda I)^\dagger z_\lambda - \delta^2 \\ & = & z_\lambda^t z_\lambda - \delta^2. \end{array}$$

Consider the case where optimality is attained at a stationary point of ν, as in Example 1.3.2. We must have $z_\lambda^t z_\lambda - \delta^2 = 0$, which implies that z_λ is on the boundary of the trust-region and therefore feasible. The second case, where the maximum occurs at the left of $-\lambda_1(D)$, as in Example 1.3.1, yields $\lambda^* = 0$, and $\nu'(\lambda) = z_\lambda^t z_\lambda - \delta^2 < 0$. This again implies feasibility for the corresponding primal vector. In the last, hard hard-case, as in Example 1.3.3, z_λ is strictly feasible and since the Hessian of Lagrangean is singular, we can add to z_λ a vector $\bar{z} \in \mathcal{N}(D + \lambda^* I)$ and move to the boundary of the trust-region.

In each case, z_λ (or the resulting $z_\lambda = z_\lambda + \bar{z}$) is feasible for the primal and $\lambda^*(z_\lambda^t z_\lambda - \delta^2) = 0$, which yields

$$\nu^* = \nu(\lambda^*) = \mathcal{L}(z_\lambda, \lambda^*) = z_\lambda^t D z_\lambda + 2c^t z_\lambda \geq \min\{z^t D z + 2c^t z\} = \mu^*.$$

This reverses the weak duality inequality. We therefore have equality and, at optimality for the dual, the corresponding primal variable, being feasible, must be optimal. □

1.3 Classification of instances

Before we embark on solution methods for *TRS*, it is instructive to look at some examples. Given the strong duality of the problem, both the primal and dual are enlightening. In the usual taxonomy of *TRS*, we distinguish three cases, most easily understood from the dual objective, expressed as

$$\nu(\lambda) = \sum_{i=1}^{n} \frac{-c_i^2}{\lambda_i + \lambda} - \lambda\delta^2.$$

As we have seen, this function is well-behaved except, possibly, when $-\lambda$ tends to one of the eigenvalues.

The first, *explicitly convex* case occurs when $D \succeq 0$. If the unconstrained minimum happens to fall within the feasible region, then the dual program is optimal at $\lambda = 0$ and the primal has solution $z_i^* = -c_i/\lambda_i$ for $1 \leq i \leq n$. This is illustrated by Example 1.3.1.

In the graphs of Example 1.3.1, and of the following examples of this section, the contour lines of the primal objective function, $\mu(x)$, are superimposed on the trust-region in the left-hand side.

The circled cross marks the optimal solution. The right-hand side graph is of the dual objective, $\nu(\lambda)$ and again, the darker point marks the optimal solution.

Example 1.3.1 Explicitly convex case. *Consider* $\min\left\{x^tQx + 2b^tx \mid x^tx \le \delta^2\right\}$,

$$Q = \begin{bmatrix} 5 & 0 \\ 0 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \delta = 1, \quad \lambda = 0, \quad x^* = \begin{bmatrix} -\frac{1}{5} \\ -\frac{1}{3} \end{bmatrix}.$$

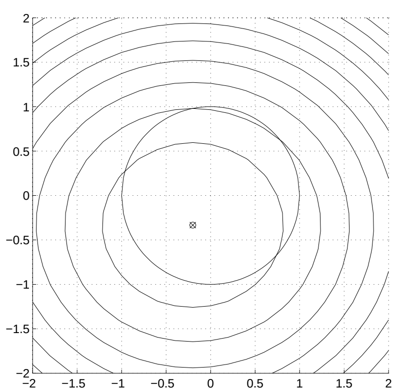


Figure 1.1: The contour lines of the primal objective function of Example 1.3.1. The trust-region is superimposed, and the dark point marks the unconstrained minimum.

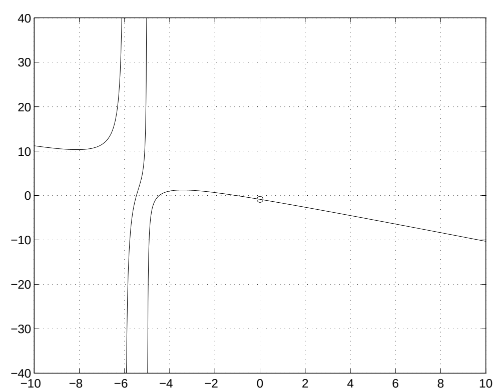


Figure 1.2: The corresponding dual objective function, $\nu(\lambda)$. *The dark point marks the optimal solution,* $\lambda = 0$ *on the domain of interest,* $\lambda \ge \max\{0, -5\}$.

This first case, if simple because the primal objective is convex, must nevertheless be handled correctly by any general-purpose *TRS* algorithm if for no other reason than that the last few iterations of most restricted-step algorithms, for which *TRS* is a subproblem, expect the trust-region not to be binding.

If the objective Hessian is singular, then the component of the primal optimal solution corresponding to the zero eigenvalues can be chosen arbitrarily to satisfy feasibility.

The convex case cannot be solved with $\lambda = 0$ if the unconstrained minimum falls outside the trust-region for then the corresponding primal value is infeasible. Nevertheless, the dual problem, a univariate maximization of a continuously differentiable function, is easy and identical to the next case.

To consider the other cases, let I be the index set of eigenvalues equal to the smallest one, $\lambda_1(Q) < 0$, which we may assume to be negative since we are no longer considering convex cases. The so-called *easy* case occurs whenever at least one $c_i \neq 0$, for $i \in I$ or, equivalently, b is not orthogonal to all eigenvectors corresponding to $\lambda_1(Q)$. Then, for the dual functional to be finite, we need $\lambda > -\lambda_1(Q)$. This is exemplified by the following where the optimal primal solution is obtained from the optimal dual λ by $z_i^* = \frac{-c_i}{\lambda_i+\lambda}$.

Example 1.3.2 Easy case. *Consider* $\min\left\{x^tQx + 2b^tx \mid x^tx \leq \delta^2\right\}$,

$$Q = \begin{bmatrix} -5 & 0 \\ 0 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \delta = 1, \quad \lambda = 6, \quad x^* = \begin{bmatrix} -1 \\ -.0833 \end{bmatrix}.$$

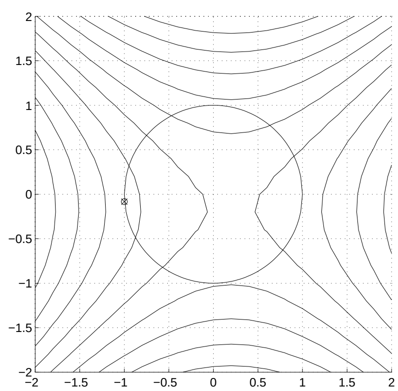


Figure 1.3: The contour lines of the primal objective function of Example 1.3.2. The trust-region is superimposed, and the dark point marks the constrained minimum.

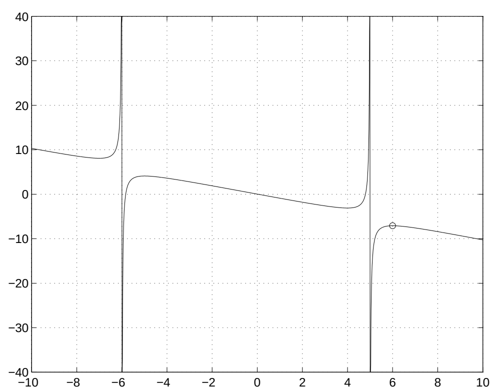


Figure 1.4: The corresponding dual objective function, $\nu(\lambda)$*. The dark point marks the optimal solution,* $\lambda = 6$ *on the domain of interest,* $\lambda \geq \max\{0, 5\}$.

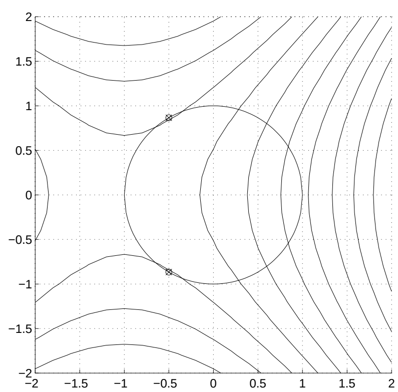
Note that the discontinuity of the dual functional at $\lambda = -\lambda_1(Q)$ requires any dual-type algorithm to stay clear of this boundary, a difficult task as the optimal solution can be arbitrarily close.

Finally, there is the so-called *hard* case where b is orthogonal to the eigenvectors corresponding to $\lambda_i, i \in I$. Then, simply enough, $\lambda = \lambda_1$. Therefore, finding the smallest eigenvalue of Q by a Lanczos algorithm, or even an approximation to it, as in the Rendl and Wolkowicz [57]) approach, goes a long way towards solving *TRS*. The hard case is exemplified by the following, where one possible optimal solution is given by

$$z_i^* = \begin{cases} \frac{-c_i}{\lambda_i+\lambda} & \text{for } i \in I := \{i|\lambda_i \neq \lambda\}; \\ \frac{\delta^2 - \sum_I z_j^* z_j}{n-|I|} & \text{otherwise.} \end{cases}$$

Example 1.3.3 Hard case. *Consider* $\min\left\{x^tQx + 2b^tx \mid x^tx \leq \delta^2\right\}$,

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \delta = 1, \quad \lambda = 1, \quad x^* = \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}.$$



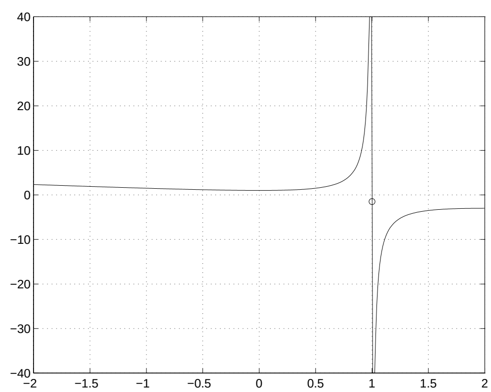


Figure 1.5: The contour lines of the primal objective function of Example 1.3.3. The trust-region is superimposed, and the dark points marks the constrained minima.

Figure 1.6: The corresponding dual objective function, $\nu(\lambda)$*. The dark point marks the optimal solution,* $\lambda = 1$ *on the domain of interest,* $\lambda \geq \max\{0, -1\}$.

In this last case, given the optimal λ, we can solve for some of the components of the optimal primal solution by $z_i^* = \frac{-c_i}{\lambda_i+\lambda}$ but this solution is not optimal yet since complementarity fails. We can nevertheless start from this point and move to the boundary in the nullspace of $Q + \lambda I$. This approach, suggested by Moré and Sorensen [44], will be discussed in the next section.

1.4 Semidefinite relaxation

Having a complete characterization of the optimality conditions of *TRS*, we can now develop an algorithm for its solution, keeping in mind that the approach must generalize, somehow, to multiple trust-regions.

The equivalence of the semidefinite and Lagrangean relaxations is our starting point. This can be seen, following the recipe developed by Poljak, Rendl and Wolkowicz [52], by first revealing the hidden semidefinite constraint, as we did in the derivation of strong duality. Taking the dual of this dual, we get a relaxation of *TRS*. But a direct approach yields the same pair of programs and highlights how the rank of the solution is tied to optimality. Recall the primal problem,

$$\textit{TRS} \qquad \min\left\{\mu(x) = x^tQx + 2b^tx \mid x^tx \le \delta^2, x \in \mathbb{R}^n\right\}.$$

To get a pure quadratic form, we need to increase the dimension of the problem by the transformation

$$y_i = x_i x_0, \quad 1 \le i \le n, \qquad y_0^2 = x_0^2 = 1.$$

The last equation serves as a normalizing condition and ensures the same optimal values in both the original and the homogenized program. The original optimal solution can be retrieved by $x_i = y_i/x_0$. We now have

$$\min\left\{\mu(x) = y^tPy \mid y^ty \le \delta^2 + 1, y_0^2 = 1, y \in \mathbb{R}^{n+1}\right\},$$

where the $(n+1) \times (n+1)$ matrix P is constructed from the original data by

$$P = \begin{bmatrix} 0 & b^t \\ b & Q \end{bmatrix}.$$

We now simplify the notation by considering

$$y^tPy = \mathrm{trace}(y^tPy) = \mathrm{trace}(Pyy^t) = \langle P, Y\rangle, \text{ for } Y = yy^t,$$

and where $\langle P, Y\rangle$ is the usual inner product of symmetric matrices. The matrix Y, as defined, is clearly positive semidefinite. We can therefore obtain a relaxation by discarding the rank one condition. The primal semidefinite program will now read

$$\mathit{PSDP} \qquad \min\left\{\tilde{\mu}(Y) = \langle P, Y\rangle \mid \langle E_{00}, Y\rangle = 1, \langle P_I, Y\rangle \leq \delta^2, Y \succeq 0\right\},$$

where

$$P = \begin{bmatrix} 0 & b^t \\ b & Q \end{bmatrix}, P_I = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, E_{00} = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{0} \end{bmatrix}, Y = \begin{bmatrix} 1 & x^t \\ x & X \end{bmatrix},$$

and all matrices are in $\mathbb{R}^{(n+1)\times(n+1)}$. The corresponding dual program is

$$\mathit{DSDP} \qquad \max\left\{\tilde{\nu}(\mu, \lambda) = -\mu - \lambda\delta^2 \mid P + \lambda P_I + \mu E_{00} \succeq 0, \lambda \geq 0\right\},$$

which we recognize as the Lagrangean dual, after homogenization, with an explicit semidefinite constraint. Note again that *PSDP*, with an additional rank one constraint on Y is exactly *TRS* since rank one implies $X = xx^t$, from which we can retrieve the solution to the original problem.

The notation $\tilde{\mu}(Y)$, if it seems abusive, since we have used $\mu(x)$ to denote *TRS*, is nevertheless justified since both optimal values are equal as we now show.

Lemma 1.4.1 *The optimal values of TRS and PSDP, denoted here respectively by $\mu(x^*)$ and $\tilde{\mu}(Y^*)$, are equal.*

Proof: Since *PSDP* is a relaxation of *TRS*, $\tilde{\mu}(Y^*) \leq \mu(x^*) = \mu^*$. Since *DSDP* is identical to the explicit dual of *TRS*, $\tilde{\nu}^* = \nu^*$, and weak duality of the semidefinite pair yields $\nu^* \leq \tilde{\mu}(Y^*)$. We therefore have the sequence of inequalities

$$\nu^* \leq \tilde{\mu}(Y^*) \leq \mu(x^*) = \mu^*.$$

Strong duality of *TRS*, expressed in the current notation as $\nu^* = \mu^*$, yields the desired result. □

If we solve the semidefinite relaxation, we therefore obtain the optimal value of *TRS*. But that is only one of the useful results of this approach as the following few lemmata will show. In the remaining development, the values of P, P_I, Y and E_{00} are defined as they were above for *PSDP*.

The first result relates the feasible set of the the semidefinite relaxation to the feasible set of *TRS*. This is a result of fundamental importance to our work. We have lifted a problem from $\mathbb{R}^n$ to $\mathbb{S}_{n+1}$. For this procedure to yield anything useful we must find a way back to the original space. In some applications of semidefinite programming, the diagonal of the optimal matrix is the key to the original problem. In our case, the first column plays this role.

The dependence we derive between feasible sets is related to a recent result of Fujie and Kojima [23] for linear objective functions over quadratically constrained sets. Our result is similar but the approach is completely different. We need the following definitions for the feasible sets involved.

$$\begin{aligned} F &:= \{x \in \mathbb{R}^n; x^t x \leq \delta^2\} \\ \tilde{F} &:= \{Y \in \mathbb{S}_{n+1}; \langle E_{00}, Y\rangle = 1, \langle P_I, Y\rangle \leq \delta^2\}, \end{aligned}$$

and a pair of maps from one space to the other,

$$\begin{aligned} P_R : \mathbb{S}_{n+1} \Rightarrow \mathbb{R}^n \quad &; \quad P_R\left(\begin{bmatrix} a & x^t \\ x & X \end{bmatrix}\right) = x, \\ P_R^{-1} : \mathbb{R}^n \Rightarrow \mathbb{S}_{n+1} \quad &; \quad P_R^{-1}(x) = \begin{bmatrix} 1 & x^t \\ x & xx^t \end{bmatrix}. \end{aligned}$$

The map P_R is a projector from Y to its first column, from which the first component is then discarded to come back to $\mathbb{R}^n$. It is invertible only when the matrix Y is rank one and its top left component is a 1.

We can now succinctly express the relation between the feasible sets.

Lemma 1.4.2 *The feasible set of TRS, denoted F, and the mapping, under P_R, of the feasible set of PSDP, denoted $\tilde{F}$ are equal. In short, $F = P_R(\tilde{F})$.*

Proof: We first show that $F \subseteq P_R(\tilde{F})$. Say $x \in F$ and let $Y = P_R^{-1}(x)$. By construction, $\langle E_{00}, Y \rangle = 1$ and, since $x^t x \leq \delta^2$,

$$\langle P_I, Y \rangle = \langle I, xx^t \rangle = x^t x \leq \delta^2.$$

Therefore, $Y \in \tilde{F}$.

In the other direction, we now show that $P_R(\tilde{F}) \subseteq F$. Say $Y \in \tilde{F}$ and let $x = P_R(Y)$, $y = (1, x)^t$. From the definition of P_I, we have the equalities $\langle P_I, Y \rangle = \langle I, X \rangle$ and $y^t P_I y = x^t x = \langle I, xx^t \rangle$. Subtracting these two equations, we get

$$\begin{aligned} y^t P_I y &= \langle P_I, Y \rangle + \langle I, xx^t \rangle - \langle I, X \rangle \\ &= \langle P_I, Y \rangle - \langle I, X - xx^t \rangle. \end{aligned}$$

Now, since $Y \succeq 0$, then X is also positive semidefinite and $X - xx^t \succeq 0$. Therefore we obtain,

$$x^t x = y^t P_I y \leq \langle P_I, Y \rangle \leq \delta^2.$$

The last inequality implies that $x \in F$. Combining the two inclusions we obtain $F = P_R(\tilde{F})$. □

This is a fairly surprising result. It provides us with a way to get feasible points to *TRS* from feasible points of the relaxation, even when these are not rank one. But a feasible pair Y, λ to the semidefinite relaxation, if Y is not rank one, will in general map to a vector x for which complementarity fails since $x^t x < \delta^2$. Interestingly, there always is a rank one solution. This follows from a result of Pataki [49] relating the number of constraints to a bound on the rank of the solution. In the semidefinite relaxation of *TRS*, the number of constraints implies a rank one solution.

Finally, note that this analysis never involves the objective function which, therefore, need not be linear or even convex. This is in contrast to the result of Fujie and Kojima [23] for linear objective over quadratic constraints.

Solving the semidefinite relaxation always yields a feasible solution to the original problem by the projection on the first column. This is interesting in itself but begs the question of optimality. The following clarifies some cases where the relaxation yields the optimal vector.

Lemma 1.4.3 *If $Q \succeq 0$, the convex case, then from Y, an optimal solution of PSDP, we can obtain $x = P_R(Y)$, optimal for TRS. Moreover, if the objective function is strictly convex, then Y is rank one.*

Proof: Say Y is optimal for *PSDP*. By the previous definition of P and of $y = (1\ x)^t$,

$$\begin{aligned} y^t P y &= 2b^t x + \langle Q, xx^t \rangle, \\ \langle P, Y \rangle &= 2b^t x + \langle Q, X \rangle. \end{aligned}$$

Subtraction of the above equations yields

$$y^t P y = \langle P, Y \rangle - \langle Q, X - xx^t \rangle \leq \langle P, Y \rangle,$$

where the inequality follows from $\langle Q, X - xx^t \rangle \geq 0$ for $Q \succeq 0$. But *PSDP* is a relaxation of *TRS*, a minimization program, therefore

$$\langle P, Y \rangle \leq y^t P y = \langle P, Y \rangle - \langle Q, X - xx^t \rangle \leq \langle P, Y \rangle,$$

and we can conclude that the inequalities are actually equalities.

If we consider now the structure of Y, we get

$$\begin{aligned} \langle P, Y \rangle - \langle Q, X - xx^t \rangle &= 2b^t x + \langle Q, X \rangle - \langle Q, X - xx^t \rangle \\ &= 2b^t x + x^t Q x. \end{aligned}$$

So that the objective values are equal, i.e.,

$$y^t P y = x^t Q x + 2b^t x = \langle P, Y \rangle.$$

Since $x = P_R(Y)$ is feasible for *TRS* by Lemma 1.4.2 and since primal and dual objective values are equal we conclude that x is optimal for *TRS*.

Moreover, since $\langle P, Y\rangle - \langle Q, X - xx^t\rangle = \langle P, Y\rangle$, we have

$$\langle Q, X - xx^t\rangle = 0.$$

If the objective is strictly convex, i.e., Q is non-singular, then $X - xx^t = 0$ and, therefore, X is rank one. □

This implies that the hard case is not a concern for convex programs.

The convex case, necessary in any general-purpose algorithm, is not the only case that the semidefinite relaxation solves naturally. We stated above and now give a formal proof that the semidefinite primal program relaxes only the rank condition.

Lemma 1.4.4 *If Y, the optimal solution of PSDP is rank one, then $x = P_R(Y)$ solves TRS.*

Proof: The argument used in Lemma 1.4.3 applies, and $y^t P y = \langle P, Y\rangle - \langle Q, X - xx^t\rangle$. If Y is rank one, then the second term vanishes and since $\langle P, Y\rangle$ is a lower bound, we get $y^t P y = \langle P, Y\rangle$ or $x = P_R(Y)$ is optimal for *TRS*. □

Unfortunately, we do not expect the relaxation to yield an optimal solution for every program. The following hints at what can go wrong and what must be done to fix it.

Lemma 1.4.5 *Assume that Y is the semidefinite optimal primal solution and that the multipliers μ, λ are optimal for the semidefinite dual program. The projection $x = P_R(Y)$, together with λ then satisfy stationarity of TRS, namely $(Q + \lambda I)x + b = 0$.*

Proof: Complementarity for the semidefinite pair yields

$$0 = \langle P + \lambda P_I + \mu E_{00}, Y\rangle.$$

But, as $P + \lambda P_I + \mu E_{00} \succeq 0$ and $Y \succeq 0$, by feasibility of *DSDP*, we must have that the matrix product is itself zero, i.e.,

$$\begin{aligned} 0 &= (P + \lambda P_I + \mu E_{00})Y \\ &= \begin{bmatrix} \mu & b^t \\ b & Q + \lambda I \end{bmatrix} \begin{bmatrix} 1 & x^t \\ x & X \end{bmatrix}. \end{aligned}$$

Therefore, considering only part of the above multiplication, $0 = b + (Q + \lambda I)x$. □

We now recognize that solving the semidefinite relaxation yields primal and dual solutions that satisfy all the necessary conditions of *TRS* with the possible exception of complementarity. Primal feasibility is shown by Lemma 1.4.2. Dual feasibility is explicit since *DSDP* is identical to the Lagrangean dual of *TRS*. Stationarity is shown in Lemma 1.4.5. We now add the last step in the semidefinite recipe to solve *TRS*.

Lemma 1.4.6 *Assume that the multipliers (μ, λ) and the matrix Y are optimal for, respectively, the primal and the dual semidefinite programs, PSDP and DSDP. Assume, moreover, that Y is not rank one. The optimal solution x^* to TRS can then be obtained by $x^* = x + \bar{x}$, where $\bar{x}$ is chosen in $\mathcal{N}(P + \lambda P_I + \mu E_{00})$, the nullspace of the Hessian of the Lagrangean, and constructed to satisfy complementarity. Since $\lambda \neq 0$, complementarity implies $(x + \bar{x})^t(x + \bar{x}) = \delta^2$.*

Proof: By Lemma 1.4.2, x is primal feasible for *TRS*. By Lemma 1.4.5, λ is dual feasible and stationarity holds. Moreover the optimal value of *TRS* is attained by Lemma 1.4.1. Only complementarity fails. If we choose $\bar{x}$ in the nullspace of $P + \lambda P_I + \mu E_{00}$ to satisfy this last requirement and therefore all conditions of Lemma 1.1.1, we conclude that $x^* = x + \bar{x}$ is optimal for *TRS*. □

The nullspace-restricted step, $\bar{x}$, can be found by a QR decomposition of the Hessian. Say $Q + \lambda I = WR$, where $W^tW = I$, then the nullspace is the solution of

$$0 = (Q + \lambda I)x,$$

which we simplify by multiplying on the left by W^t, to obtain

$$0 = W^t(Q + \lambda I)x = W^t(WR)x = Rx.$$

This equation can easily be solved by back substitution since the matrix R is upper trapezoidal.

This last ingredient in the semidefinite framework is interesting in more ways than one. As we will shortly see, it translates into a very simple algorithm for solving *TRS*. But, incidentally, it also provides us with another, arguably simpler proof, that *TRS* has no duality gap. We have shown this result before but repeat it for the simplicity of the proof.

Lemma 1.2.1 (revisited) *Strong duality holds for TRS.*

Proof: The semidefinite dual pair *PSDP*, *DSDP* has strictly interior points and therefore no duality gap, i.e., $\tilde{\nu}^* = \tilde{\mu}^*$. By construction, the dual program *DSDP* is equivalent to the Lagrangean relaxation of *TRS*. Therefore $\tilde{\nu}^* = \nu^*$, both dual optimal values are equal. By the previous lemmata about solving *PSDP*, in cases where the solution is rank one (Lemma 1.4.4) and in cases where it is not (Lemma 1.4.6), we can find primal solutions X and x such that $\tilde{\mu}^* = \mu^*$. Therefore $\mu^* = \nu^*$; the primal and dual optimal values of *TRS* are attained and are equal. □

We now combine these results into a very simple *TRS*-solving algorithm.

1.5 Simple solution to *TRS*

We have shown how a semidefinite framework allows us, in theory, to solve *TRS*. In the explicitly convex and easy non-convex cases, the semidefinite pair of programs yields the optimal solution with no additional work. In the hard case, a decomposition of the Hessian of the singular Lagrangean is required to move to the boundary. The following pseudo-code description makes this somewhat more explicit.

The full statement of the algorithm makes painfully clear its simplicity. But we view this simplicity as a strength of the theoretical background and of the semidefinite framework, not as a weakness. We have yet to see how we can, in practice, solve the semidefinite relaxation of

TRS. But as our intentions, in this work, do not involve studying the intricacies of interior-point algorithms, we prefer to consider the interior-point code simply as a tool. Moreover, as this relaxation of *TRS* is only a special case of a more general semidefinite pair, we will postpone the interior-point algorithm discussion to a later chapter, where the general program is known.

ALGORITHM FOR TRUST-REGION SUBPROBLEM

$\boldsymbol{TRS}(Q, b, \delta^2)$

$Y \in \operatorname{argmin}\{\langle P, Y\rangle : \langle P_I, Y\rangle \leq \delta^2, \langle E_{00}, Y\rangle = 1\}$

$(\mu, \lambda) \in \operatorname{argmax}\{-\mu - \lambda\delta^2 : P + \lambda P_I + \mu E_{00} \succeq 0, \lambda \geq 0\}$

if $\lambda(x^t x - \delta^2) = 0$

$x^* = x$

else

Find $\bar{x} \in \mathcal{N}(Q + \lambda I)$ with $\bar{x}^t \bar{x} = \delta^2 - x^t x$

$x^* = x + \bar{x}$

fi

return(x^*)

1.6 A first step

The main purpose of this chapter was to introduce the semidefinite framework used to solve a fully quadratic program, where both the objective and the constraints are second-order polynomials. We have seen how a very simple semidefinite formulation solves the trust-region subproblem and how we might implement this formulation.

The trust-region subproblem can and has been generalized in a number of ways. Moré [43] has considered relaxing the positive definite requirement of the constraint and has developed algorithms to solve such problems. Stern and Wolkowicz [64] also considered indefinite trust regions, with both upper and lower bounds, of the form

$$\min\left\{\mu(x) = x^t Q x + 2b^t x \mid \delta \leq x^t D x \leq \alpha, x \in \mathbb{R}^n\right\}.$$

Recently, Di and Sun [15] reconsidered the origins of the *TRS* in the optimization world where the objective is usually a model for a more complex nonlinear function and decided to explore a conic model of the objective function. The resulting subproblem is expressed as

$$\min\left\{f + \frac{b^t x}{1 - a^t x} + \frac{1}{2}\frac{x^t Q x}{(1 - a^t x)^2} \mid x^t D x \leq \delta^2, x \in \mathbb{R}^n\right\}.$$

The direction we intend to explore next considers the same simple quadratic objective we have considered in this chapter but constrained by two, partially overlapping, trust regions.

Chapter 2

Two trust-regions

Moving up in complexity, we consider now the two trust-region problem, hereafter *2-TRS*. It occurs naturally, for example in process modeling, under the guise of the dual-response problem. (See Myers and Carter [45].) But perhaps more important for our development is its appearance as a subproblem of general nonlinear solvers for $\min\left\{f(x) \mid h(x) = 0\right\}$ that use a sequence of approximations of the type

$$CDT \qquad \min\left\{\mu(x) = x^tQx + 2b^tx \mid ||A^tx + c|| \leq \epsilon, ||x|| \leq \delta, x \in \mathbb{R}^n\right\}.$$

The problem was introduced by Celis, Dennis and Tapia [10], hence the acronym *CDT*, under which it is now known, and is the stepping stone of a family of iterative methods. (See Byrd, Schnabel and Schultz [9], Powell and Yuan [55], Yuan [75], Williamson [73], El-Alem [18], Zhang [77].)

Because we wish a formulation more conducive to a semidefinite program and wish to handle inequalities, we transform CDT into a form more appropriate to our purpose. We first square the constraints to get

$$\min\left\{\mu(x) = x^tQx + 2b^tx \mid x^tAA^tx + 2c^tA^tx + c^tc \leq \epsilon^2, x^tx \leq \delta^2, x \in \mathbb{R}^n\right\},$$

and now rewrite to introduce the notation we will use throughout,

$$\textit{2-TRS} \qquad \min\left\{\mu(x) = x^tQx + 2b^tx \mid x^tQ_1x + 2b_1^tx - a_1 \leq 0, x^tx \leq \delta^2, x \in \mathbb{R}^n\right\}.$$

For Q_1 positive definite, as we will assume in this chapter, *2-TRS* is equivalent to *CDT* and is clearly a generalization of *TRS*. Lifting the restriction on Q_1, in the manner of Moré [43], possibly with the additional complexity of a two-sided trust-region, in the manner of Stern and Wolkowicz [63], yields a more general program we might also wish to consider.

We impose no restriction on the objective function. Heinkenschlos [28] studied programs resulting from a convexity restriction; so have Yuan [76], developing a dual algorithm for their solution, and Zhang [77], with a parametric approach .

Throughout this chapter, unless otherwise stated, we assume that the feasible region has a non-empty interior. For the problem under consideration, such a constraint qualification is not overly restrictive.

2.1 Characterization of optimality

The optimality conditions were originally studied by Yuan [75], albeit in a very different manner. We wish to parallel the development of the previous chapter and therefore first investigate necessary conditions.

Throughout this chapter, references are made to the *tangency* of some constraints. By tangent constraints at x, we mean that the gradients of the constraints, evaluated at x, are linearly dependent.

Lemma 2.1.1 *If $x \in \mathbb{R}^n$ is a local solution of 2-TRS, then there exists a vector of nonnegative multipliers $\lambda \in \mathbb{R}^2$ satisfying*

$$(Q + \lambda_1 Q_1 + \lambda_2 I)x = -b - \lambda_1 Q_1 \qquad \textit{(stationarity)},$$

$$\left.\begin{array}{l} \lambda_1(x^tQ_1x + 2b_1^tx - a_1) = 0 \\ \lambda_2(x^tx - \delta^2) = 0 \end{array}\right\} \qquad \textit{(complementarity)},$$

and where the number of negative eigenvalues of $Q + \lambda_1 Q_1 + \lambda_2 I$ *is*

$$\left.\begin{array}{l} 0 \\ \text{at most } 1 \\ \text{at most } 2 \end{array}\right\} \ \textbf{if} \ \left\{\begin{array}{l} \text{no constraint are active at } x, \\ \text{one constraint is active at } x, \text{ and} \\ \text{both constraints are active but not tangent at } x. \end{array}\right.$$

Proof: Since we assumed the existence of strictly interior points, the standard first-order conditions yield nonnegative multipliers satisfying stationarity and complementarity. We are left to consider the Hessian of the Lagrangean,

$$\begin{aligned} \mathcal{L}(x, \lambda) &= x^t Q x + 2b^t x + \lambda_1 (x^t Q_1 x + 2b_1^t x - a_1) + \lambda_2 (x^t x - \delta^2) \\ &= x^t (Q + \lambda_1 Q_1 + \lambda_2 I) x + 2(b + \lambda_1 b_1) x - \lambda_1 a_1 - \lambda_2 \delta^2. \end{aligned}$$

In the event that x satisfies constraint i with equality, we can define the tangent plane T_i to constraint i as

$$\begin{aligned} T_1 &:= \{y : y^t v_1 = 0, v_1 = Q_1 x + b_1\} \\ T_2 &:= \{y : y^t v_2 = 0, v_2 = x\}. \end{aligned}$$

The standard necessary second-order conditions require the inequality

$$y^t (Q + \lambda_1 Q_1 + \lambda_2 I) y \geq 0, \quad \text{for all } y \text{ in the tangent space.}$$

If no constraints are active, then $Q + \lambda_1 Q_1 + \lambda_2 I$ must be positive semidefinite on the whole n-dimensional space. The Hessian has no negative eigenvalue.

If one constraint, say i, is active, the corresponding tangent plane T_i is of dimension $n - 1$ and the Hessian, by Corollary 1.1.2 to the Courant-Fisher Theorem, has at most one negative eigenvalue.

If both constraints are active and the tangent planes do not coincide, then the tangent space $T_1 \cap T_2$ is a $(n-2)$-dimensional subspace and the Hessian, again by Corollary 1.1.2, has at most two negative eigenvalues. □

The previous lemma allows the number of negative eigenvalue of the Lagrangean to be as much as two when both constraints are active and is silent about the degenerate case (when the constraints are tangent). First we note that the bound on the number of negative eigenvalues can be strengthened. A very recent result of Peng and Yuan [51] concludes that the Hessian has at most one negative eigenvalue when both constraints are active. We state the result without proof.

Proposition 2.1.2 *If both constraints are active but not tangent at an optimal x, then there exists optimal multipliers λ for which $Q + \lambda_1 Q_1 + \lambda_2 I$ has at most one negative eigenvalue.*

Peng and Yuan [51] go on to consider the degenerate case, where the gradients of the constraints at the optimal point x are linearly dependent. This degenerate case may be of little interest in practice but the result is nevertheless interesting. The multipliers are not uniquely determined and a wrong choice may lead to a Hessian which is not semidefinite. But there is a right choice of multipliers.

Proposition 2.1.3 *If both constraints are active and tangent at an optimal x, then there is a choice of optimal Lagrange multipliers λ such that $Q + \lambda_1 Q_1 + \lambda_2 I$ is positive semidefinite.*

We will not give the proof but an intuitive understanding of the result is that, when both constraints are active and tangent, the optimal solution is a global minimizer for one of the trust-regions. Said differently, one of the constraint is redundant. This is clear in the 3-dimensional case as can be seen in Example 2.3.3.

This case is the only one where the multipliers λ are not uniquely defined. This is a well-known consequence of the *strong* constraint qualification: linear independence of the gradients of the constraints. (See Bazaraa and Shetty [2] for more details on the relative strength of various constraints qualification.) We sketch a simple proof of the uniqueness of the multipliers in the general case.

Lemma 2.1.4 *Consider the program $\min\left\{f(x) \;\middle|\; g(x) \leq 0, x \in \mathbb{R}^n\right\}$. If we assume that x is a local optimum and that the gradients of the active constraints are linearly independent at x, then the optimal Lagrange multipliers corresponding to this primal solution are uniquely defined.*

Proof: Optimal multipliers satisfy stationarity of the Lagrangean, a necessary condition at the local optimum x,

$$\nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) = 0,$$

or

$$[\nabla g_1(x) \ldots \nabla g_m(x)] \begin{bmatrix} \lambda_1 \\ \cdots \\ \lambda_m \end{bmatrix} = -\nabla f(x).$$

Linear independence of the gradients of the constraints at x imply that the above matrix is full column rank and the system, therefore, has a unique solution, which can be explicitly written as

$$\lambda = -\left[\nabla g_1(x) \ldots \nabla g_m(x)\right]^{\dagger} \nabla f(x),$$

where $(\cdot)^{\dagger}$ is the Moore-Penrose inverse. □

The generalized inverse is a nice expression for the solution of the linear system but numerically it may be more efficient to find a decomposition $[\nabla g_1(x) \ldots \nabla g_m(x)] = QR$, where Q is orthogonal and then solve by back substitution $R\lambda = -Q^t \nabla f(x)$.

In the hierarchy of constraint qualifications, linear independence of the gradients is one of the more restrictive. But in the context of *2-TRS*, where one of the trust-regions is usually under our control, it is reasonable to expect the trust-regions not to be tangent in the generic case. This allows us to usefully restrict the previous result to the *2-TRS* case.

Lemma 2.1.5 *Unless both constraints are active and tangent at an optimal solution x, the multipliers λ_1 and λ_2, corresponding to x, are uniquely defined.*

Proof: Non-tangent constraints imply linear independence of the gradients. The result follows from Lemma 2.1.4. □

As opposed to *TRS*, the program *2-TRS* does not exhibit any hidden convexity. We cannot, therefore, expect the necessary and sufficient conditions to coincide. A gap may remain between the primal and dual optimal values, as we will see in the next section. There are nonetheless sufficient Lagrange conditions that have been known for a long time. (See Luenberger [36], p223.)

Lemma 2.1.6 *Assume that x^* is feasible for the program*

$$\text{NLP} \qquad \min\Big\{f(x) \mid g(x) \leq 0, x \in \mathbb{R}^n\Big\}.$$

Assume also that there is a nonnegative vector $\lambda^* \in \mathbb{R}^m$ such that the pair x^*, λ^* satisfies stationarity and complementarity. Finally, assume that the Lagrangean, $\mathcal{L}(x, \lambda^*)$ is convex. Then x^* is optimal for NLP. Moreover, x^* is the unique optimal solution if the Lagrangean is strictly convex.

Proof: By hypothesis, the program

$$\min\Big\{\mathcal{L}(x, \lambda^*) = f(x) + g(x)^t\lambda^* \mid x \in \mathbb{R}^n\Big\}$$

is convex and stationarity, which is assumed to hold at (x^*, λ^*), implies that

$$\mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*), \quad \text{for all} \quad x \in \mathbb{R}^n.$$

If we consider the left-hand side of this inequality, we can simplify to $\mathcal{L}(x^*, \lambda^*) = f(x^*) + g(x^*)^t\lambda^* = f(x^*)$ by complementarity. On the other hand, if we restrict ourselves to the set of x feasible for *NLP*, since $\lambda^* \geq 0$ and $g(x) \leq 0$, we obtain $\mathcal{L}(x, \lambda^*) = f(x) + g(x)^t\lambda^* \leq f(x)$. We can now conclude from our original inequality that

$$f(x^*) = \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*) \leq f(x),$$

or that x^* is optimal for *NLP*.

If the Hessian of the Lagrangean is positive definite, the first inequality now reads

$$\mathcal{L}(x^*, \lambda^*) < \mathcal{L}(x, \lambda^*), \quad \text{for all} \quad x \in \mathbb{R}^n,$$

and we conclude that the optimal solution is unique. □

This result, restricted to *2-TRS*, can be expressed somewhat more compactly because the

Hessian does not depend on x. (For a different proof, see Zhang [77].)

Lemma 2.1.7 *Assume that a vector x, feasible for 2-TRS, together with a nonnegative vector $\lambda \in \mathbb{R}^2$ satisfy stationarity, complementarity and $Q + \lambda_1 Q_1 + \lambda_2 I \succeq 0$. Then x is optimal for 2-TRS. Moreover, if $Q + \lambda_1 Q_1 + \lambda_2 I \succ 0$, then x is the unique optimum.*

Proof: The result follows from Lagrangean sufficiency, Lemma 2.1.6. □

The above proof holds for any number of active constraints at optimality but it is instructive to consider the case of a single active constraint separately. Assuming $\lambda_1 = 0$, for example, the conditions of Lemma 2.1.7 reduce to

$$(Q + \lambda_2 I)x = -b, \qquad \lambda_2(x^t x - \delta^2) = 0, \qquad Q + \lambda_2 I \succeq 0,$$

conditions we recognize as necessary and sufficient for the optimality of *TRS*. This occurs when the objective function is convex or, more generally, when one constraint isolates the global minimum of the implicit *TRS* forced by the other constraint. In this manner, we relate *2-TRS* to our previous considerations of *TRS*.

2.2 Lagrangean relaxation

We have hinted at a gap between primal and dual optimal values of *2-TRS*. It is time we describe the dual program and show where and why these gaps originate.

To derive the dual, we first state *2-TRS* as a minimax program,

$$\min\Big\{\max\left\{x^t Q x + 2b^t x + \lambda_1(x^t Q_1 x + 2b_1^t x - a_1) + \lambda_2(x^t x - \delta^2) \mid \lambda \geq 0\right\} \;\Big|\; x \in \mathbb{R}^n\Big\}.$$

As noted above, we can homogenize the quadratic function by $y = (x_0 \ x)^t$, yet obtain the same optimal value if we require $x_0^2 = 1$. The homogenized program reads

$$\min\Big\{\max\left\{y^t P y + y^t \lambda_1 P_1 y + y^t \lambda_2 P_2 y + y^t \mu E_{00} y - \mu \mid \lambda \geq 0, \mu \in \mathbb{R}\right\} \;\Big|\; y \in \mathbb{R}^{n+1}\Big\},$$

where

$$P = \begin{bmatrix} 0 & b^t \\ b & Q \end{bmatrix}, P_1 = \begin{bmatrix} -a_1 & b_1^t \\ b_1 & Q_1 \end{bmatrix}, P_2 = \begin{bmatrix} -\delta^2 & 0 \\ 0 & I \end{bmatrix}, E_{00} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

To see the equivalence of this formulation with our previous exposition of *2-TRS*, we note that for the inner maximization to be bounded, we must have

$$y^t P_1 y \leq 0, \qquad y^t P_2 y \leq 0, \quad \text{and} \quad y^t \mu E_{00} y = \mu.$$

These are equivalent to

$$x^t Q_1 x + 2b_1^t x - a_1 \leq 0, \qquad x^t x - \delta^2 \leq 0, \quad \text{and} \quad y_0^2 = 1,$$

conditions we recognize as primal feasibility for the homogenized *2-TRS*. The maximization will be attained when

$$x^t Q_1 x + 2b_1^t x - a_1 = 0 \quad \text{and} \quad x^t x - \delta^2 = 0,$$

and we are left with the primal objective function to minimize.

To derive the dual, we interchange min and max and rearrange the Lagrangean to read

$$\max \Big\{ \min \left\{ y^t (P + \lambda_1 P_1 + \lambda_2 P_2 + \mu E_{00}) y - \mu \mid y \in \mathbb{R}^{n+1} \right\} \bigm| \lambda \geq 0, \mu \in \mathbb{R} \Big\}.$$

For the inner minimization to be bounded we must now have

$$P + \lambda_1 P_1 + \lambda_2 P_2 + \mu E_{00} \succeq 0.$$

Since all principal minors of a positive semidefinite matrix are positive semidefinite, this implies

$$Q + \lambda_1 Q_1 + \lambda_2 I \succeq 0.$$

This is where the duality gap arises. We remember that the optimality conditions, for *2-TRS*, did not require the Hessian of the Lagrangean to be semidefinite. It was allowed up to two negative eigenvalues. But the Lagrangean dual program we are deriving here requires the same Hessian to be semidefinite. We therefore cannot expect the primal variables corresponding to an optimal dual solution to be optimal for *2-TRS*. They will be optimal only in cases where the Lagrangean is convex at primal optimality.

We now pursue the derivation and note that the minimum over x will be attained at $x = 0$ from which we get the dual program

$$\textit{D2-TRS} \qquad \max\left\{-\lambda_1 a_1 - \lambda_2 \delta^2 - \mu \mid P + \lambda_1 P_1 + \lambda_2 P_2 + \mu E_{00} \succeq 0, \lambda \geq 0\right\}.$$

This is a semidefinite program of low dimension, whatever the original dimension of *2-TRS*, and is therefore potentially easier to solve. But unfortunately, as we will see by examples in the next section, solving the dual does not, in general, yield an optimal primal solution.

There is a sense in which we can eradicate the duality gap by transforming *2-TRS*. The transformation involves adding a constraint.

Lemma 2.2.1 *Suppose that x is a primal optimal solution to 2-TRS with associated Lagrange multipliers λ_1 and λ_2. Then there exists a quadratic constraint that, added to the problem, will yield a convex Hessian while retaining x as an optimal solution.*

Proof: First we find the required multiplier λ_3 such that

$$Q + \lambda_1 Q_1 + \lambda_2 I + \lambda_3 I \succeq 0.$$

Such a multiplier is clearly not unique but we can make it so by choosing

$$\lambda_3 = \max\left\{0, -\lambda_{\min}(Q + \lambda_1 Q_1 + \lambda_2 I)\right\}.$$

If $\lambda_3 = 0$ we are done, as the Lagrangean is convex. If not, we choose a vector b_3 so that stationarity is still satisfied, namely,

$$(Q + \lambda_1 Q_1 + \lambda_2 I + \lambda_3 I)x + (b + \lambda_1 b_1 + \lambda_3 b_3) = 0.$$

This is always possible. In a pinch, $b_3 = x$ will do. And finally, since we forced λ_3, to be positive the third constraint must be binding and we compute a_3 such that

$$x^t x + 2b_3^t x + a_3 = 0.$$

This is again always possible. By construction, all sufficient conditions for optimality (Lemma 2.1.7) are satisfied. Therefore x remains optimal. □

If we have chosen $b_3 = x$, then $a_3 = 0$ and the new constraint has no relative interior. This is somewhat uninteresting. But better choices for b_3 do exist. And it may be possible to find them in a consistent and generally applicable manner that would lead to an algorithm. The question remains open.

The lemma does provide a *necessary condition* for optimality, the driving force behind most algorithms: At some iterate, potentially optimal, test necessary conditions and act on a failure in order to improve the iterate. In the *2-TRS* case, it might be possible, for example, to add a constraint of maximum volume.

2.3 Classification of instances

In the following examples we wish to highlight the different cases described by the necessary conditions of Lemma 2.1.1. First we want to show that they can all occur and that, therefore, there are real gaps between primal and dual optimal values. But we also wish to consider the relation between the optimal solution of *2-TRS* and our previous work on the single trust-region problem.

Since the necessary conditions of *2-TRS* fall naturally into three cases, according to the number of active constraints, the examples will fall into the same classes. We skip the first case as it is not essentially different from the *TRS* case where the constraint is inactive. No active constraints implies a convex objective function and, therefore, a convex Lagrangean and no duality gap. In fact, a convex objective, wherever the unconstrained optimum falls, implies a convex Lagrangean and no duality gap.

As important as the convex case is, examples shed no additional light on the problem. In the following, we therefore consider only non-convex objective functions. The vectors x^* and λ^* denote primal and dual optimal solutions.

The cases we illustrate are

- One active constraint
  - Isolated global minimum of implicit trust-region (Example 2.3.1)
  - Isolated non-global minimum (Example 2.3.2)
- Two active constraints
  - Tangent active constraints (Example 2.3.3)
  - Intersecting active constraints (Example 2.3.4)

In light of our work on *TRS*, the first interesting case occurs when only one constraint is active at the optimal solution x^* of *2-TRS*. The necessary conditions of Lemma 2.1.1 tell us that the Hessian $Q + \lambda_1 Q_1 + \lambda_2 I$ has at most one negative eigenvalue.

We consider first the case of a convex Lagrangean. Since only one constraint is active, one of the multipliers, say λ_1 is zero and the Hessian therefore satisfies $Q + \lambda_2 I \succeq 0$ so that all necessary (Lemma 1.1.3) and sufficient (Lemma 1.1.1) conditions of *TRS* are satisfied for the binding constraint. The second constraint must isolate the global minimum implicit in the active trust-region. This is illustrated by the following.

Example 2.3.1 Isolated global minimizer.

Consider $\min\left\{x^tQx + 2b^tx \mid x^tQ_1x + 2b_1^tx - a_1 \leq 0, x^tx \leq \delta^2\right\}$,

$$Q = \begin{bmatrix} -2 & 0 \\ 0 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, b_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, a_1 = 1, \delta^2 = 4.$$

The optimal solution is $\lambda^* = \begin{bmatrix} 0 \\ 2.5 \end{bmatrix}, x^* = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$, *with* $\nabla^2\mathcal{L} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{9}{2} \end{bmatrix}$.

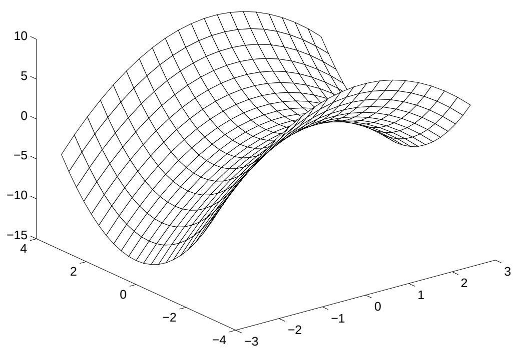


Figure 2.1: The primal objective function of Example 2.3.1.

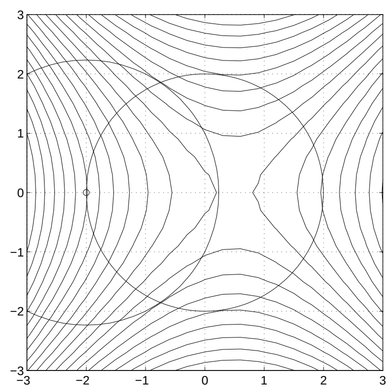


Figure 2.2: The corresponding contour lines, superimposed on the two trust regions. The dark point marks the isolated global minimum.

Note that the Hessian of the Lagrangean, in this example, is positive definite, clearly indicating a global minimum, but need only be semidefinite. Either way, there is no duality gap; the primal and dual optimal values are equal.

The second case concerns a Hessian with exactly one negative eigenvalue at x^*, the optimal solution of *2-TRS*. Again, because one multiplier is zero, the condition that the Hessian satisfies, namely $Q + \lambda_1 Q_1 + \lambda_2 I \succeq 0$, concerns only the active trust-region. Yet the negative eigenvalue implies that x^* cannot be a global minimum for this trust-region considered separately. It is rather a local, non-global minimum. The following example is a slight variation of the last one.

Example 2.3.2 Isolated non-global minimizer.

Consider $\min\left\{x^tQx + 2b^tx \mid x^tQ_1x + 2b_1^tx - a_1 \leq 0, x^tx \leq \delta^2\right\}$,

$$Q = \begin{bmatrix} -2 & 0 \\ 0 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, b_1 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, a_1 = 1, \delta^2 = 4.$$

$$\textit{The optimal solution is } \lambda^* = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, x^* = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \textit{with } \nabla^2\mathcal{L} = \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & \frac{7}{2} \end{bmatrix}.$$

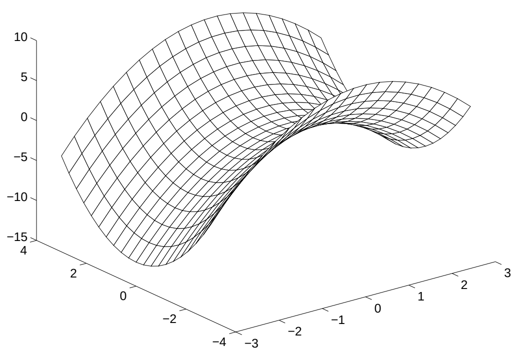


Figure 2.3: The primal objective function of Example 2.3.2, which is the same as Example 2.3.1. The difference lies in the trust-regions.

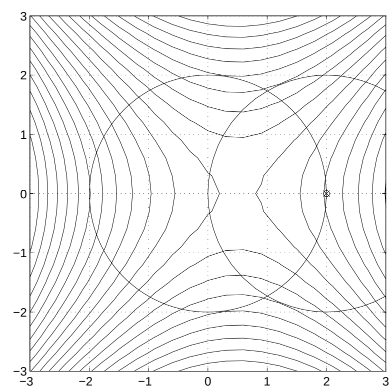


Figure 2.4: The corresponding contour lines and the two trust regions. The dark point marks the isolated local, non-global minimum.

The important point highlighted by this example is that the second constraint, even if not binding, modifies the problem in a fundamental way. Because the Lagrangean is no longer convex, there will be a gap between the primal and dual optimal values. Solving the dual is no longer sufficient but we might consider solving for local, non-global minima of each trust-regions, in turn. We have tried, somewhat informally, to gauge the frequency of such duality gaps by generating random *2-TRS* problems of various dimensions. Almost a third of the problems fell in that category, suggesting that algorithms purporting to solve *2-TRS* cannot ignore them.

Even if we rarely expect, in practice, to encounter cases where both constraints are active, they are possible. For illustrative purposes, we can ignore instances where one constraint is only *weakly active* in the sense that complementarity holds with the constraint at equality and the multiplier at zero. These are limiting cases that can be considered either from the point of view of a single active constraint or as two strongly active constraints, as we intend to do. The following example is from Yuan[75].

Example 2.3.3 Two tangent active constraints.

Consider $\min\left\{x^tQx + 2b^tx \mid x^tQ_1x + 2b_1^tx - a_1 \leq 0, x^tx \leq \delta^2\right\}$,

$$Q = \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix}, b = \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}, Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, b_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, a_1 = 0, \delta^2 = 4.$$

$$\textit{The optimal solution is } \lambda^* = \begin{bmatrix} 0 \\ \frac{7}{4} \end{bmatrix}, x^* = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \textit{with } \nabla^2\mathcal{L} = \begin{bmatrix} \frac{-1}{4} & 0 \\ 0 & \frac{-5}{4} \end{bmatrix}.$$

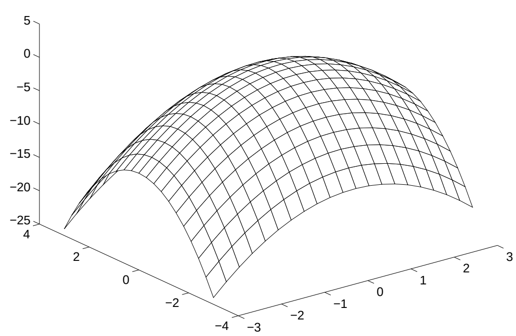


Figure 2.5: The primal objective function of Example 2.3.3.

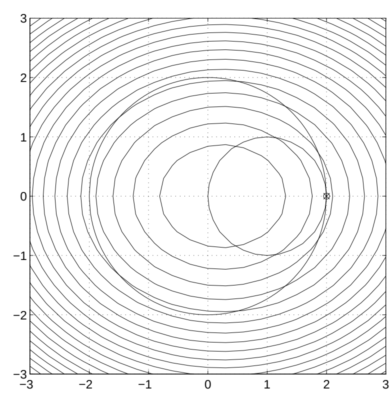


Figure 2.6: The corresponding contour lines, superimposed on the two tangent trust regions.

Both constraints can be tangent at optimal x^* so that the gradients are linearly dependent. In that case, as Lemma 2.1.5 showed, the multipliers are not uniquely defined and some valid choices will lead to a Hessian with two negative eigenvalues. In this degenerate case it is always possible, as Proposition 2.1.3 stated, to choose multipliers that will avoid Hessians with negative eigenvalues. There are therefore no duality gaps as one constraint is essentially redundant.

If the constraints are not tangent, but are *strongly* active (non-zero multipliers), then neither are redundant and the optimal solution x^* is not a local minimum of either trust-regions considered separately. The Hessian of the Lagrangean may have at most one negative eigenvalue in which case there would be a duality gap. Or it may be positive semidefinite, as the following illustrates.

Example 2.3.4 Two intersecting active constraints.

Consider $\min\left\{x^tQx + 2b^tx \mid x^tQ_1x + 2b_1^tx - a_1 \le 0, x^tx \le \delta^2\right\}$,

$$Q = \begin{bmatrix} -4 & 0 \\ 0 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, b_1 = \begin{bmatrix} 2 \\ \frac{3}{2} \end{bmatrix}, a_1 = 5, \delta^2 = 6.$$

$$\textit{The optimal solution } \lambda^* = \begin{bmatrix} 2.63 \\ 3.97 \end{bmatrix}, x^* = \begin{bmatrix} -2.41 \\ -0.46 \end{bmatrix}, \textit{with } \nabla^2\mathcal{L} = \begin{bmatrix} 2.6 & 0 \\ 0 & 8.6 \end{bmatrix}.$$

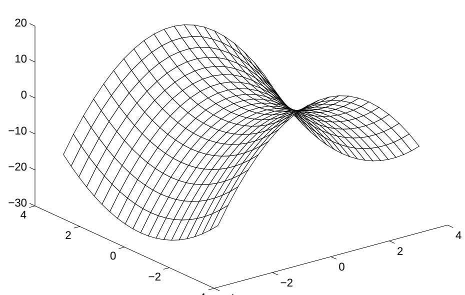


Figure 2.7: The primal objective function of Example 2.3.4.

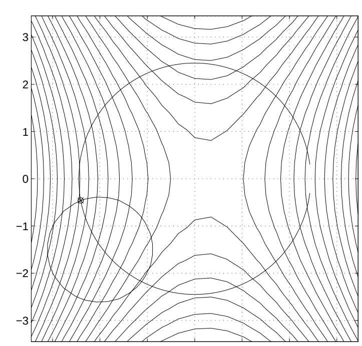


Figure 2.8: The corresponding contour lines, superimposed on the two trust regions. The dark point marks the global minimum.

Is it noteworthy that the Hessian is positive definite in this case so that we expect no duality gap but there is no reason yet to believe that this is representative of the two active constraint case.

We close our bestiary of two trust-region problems after we remark, once more, that our informal classification relied greatly on the implicit existence of local minimizers, both global and non-global of the active trust-region problem. We will try, in the following section, to analyze more closely the existence of these minimizers, for a solution approach may use these to its advantage.

2.4 Local minimizers

The existence of local minimizers deserves more attention. We pursue Example 2.3.2 above where we saw both global and non-global minimizers.

Example 2.4.1 *Example 2.3.2 revisited.*

$$\textit{Global } x^* = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, \lambda^* = \frac{5}{2}, \mu^* = \nu^* = -12, \ \textit{Non-global } x = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \lambda = \frac{3}{2}, \nu = -4.$$

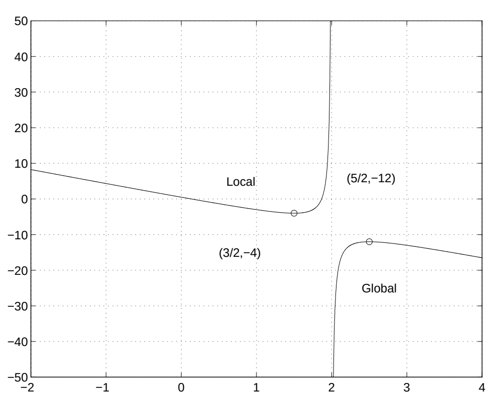


Figure 2.9: Stationary points of the dual functional, $\nu(\lambda)$, corresponding to Example 2.4.1

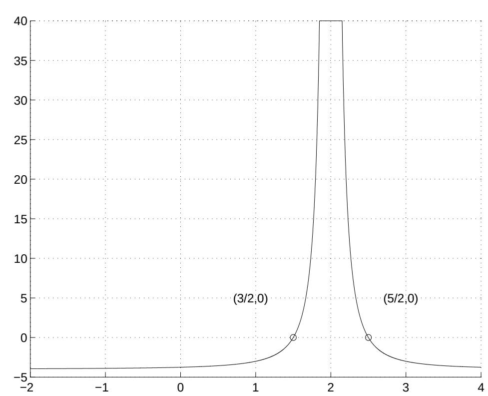


Figure 2.10: And its first derivative, $\nu'(\lambda)$.

As we have seen, the number of negative eigenvalues of the Lagrangean is often related to the presence of local minima of one trust-region within the feasible set specified by the other. How often such minima are isolated will strongly depend on the origin of the problem. In a Levenberg-Marquadt type algorithm, for example, the added trust-region is expected to be the only one active until, asymptotically, it becomes redundant. Therefore, most of the time, a global minimum will be feasible.

But in other settings, the existence of non-global minima may play an important role. As we stated in the first chapter, the existence of such non-global minima, explored by Lyle and Szularz [39] via the dual Lagrangean, was fully characterized by Martinez [42] and is a somewhat surprising result. We will not prove the result but rather provide a sketch.

Recall the diagonalized form of the single trust-region program

$$TRS \qquad \min\left\{\mu(x) = z^t Dz + 2c^t z \;\middle|\; z^t z \leq \delta^2, z \in \mathbb{R}^n\right\}.$$

With this formulation, we can state Martinez's result in the following manner.

Proposition 2.4.2 *There is at most one local, non-global minimizer x of TRS. If there is one such x, then $c_i \neq 0$ for all i where $\lambda_i = \lambda_1(D+\lambda I)$, and the dual optimal solution λ occurs in the interval $(-\lambda_2, -\lambda_1)$.*

Sketch of proof : By Lemma 1.1.3, a local optimum dual pair x, λ of TRS must satisfy $y^t(D + \lambda I)y \geq 0$, for all y such that $y^t x = 0$. From this we conclude that $D + \lambda I$ has at most one negative eigenvalue. Now, by Lemma 1.1.1, it must have exactly one such eigenvalue for x not to be global. Therefore, we can now restrict our attention to an interval where $D + \lambda I$ is full rank and invertible, $\lambda \in (-\lambda_2, -\lambda_1)$ and the pair x, λ must satisfy $x^t(D + \lambda I)x < 0$.

By stationarity, $x = -(D + \lambda I)^{-1}c$, and the above inequality can be rewritten,

$$x^t(D + \lambda I)x = c^t(D + \lambda I)^{-1}(D + \lambda I)(D + \lambda I)^{-1}c = c^t(D + \lambda I)^{-1}c < 0.$$

This provides the first characterizing condition of a local minimum. Now, since $\lambda \neq 0$, because a local minimum must be on the boundary, complementarity yields

$$\delta^2 - x^t x = \delta^2 - c^t (D + \lambda I)^{-1} (D + \lambda I)^{-1} c = \delta^2 - c^t (D + \lambda I)^{-2} c = 0,$$

the second condition.

Rewriting this pair of conditions in terms of the so-called *explicit secular* function [72],we get

$$\begin{array}{rclcl} 0 & > & \sum_{i=1}^n \frac{c_i^2}{\lambda_i + \lambda} & = & -\nu(\lambda) - \lambda\delta^2 \\ 0 & = & \sum_{i=1}^n \frac{c_i^2}{(\lambda_i + \lambda)^2} - \delta^2 & = & \nu'(\lambda). \end{array}$$

With this formulation of the necessary conditions for a non-global minimum, it is clear that there can be no more than two such points on the interval $\lambda \in (-\lambda_2, -\lambda_1)$. (See Example 2.4.1.) Not so easy to see is that there can be only one. Herein lies the surprising result of Martinez. □

With a little additional work, we can see that there will never be a non-global minimum if the global minimum is an instance of the hard case: For $\lambda_i \neq \lambda_1$, the denominators $\lambda_i + \lambda$ are all positive on the domain of interest, $\lambda \in (-\lambda_2, -\lambda_1)$. In order to satisfy the first of the two required conditions for non-global minima, we therefore must have $c_i \neq 0$ for some i such that $\lambda_i = \lambda_1(D + \lambda I)$. And since the hard case is characterized by all such $c_i = 0$, it cannot harbor non-global minima.

Martinez went on to describe an algorithm to find these non-global minima. Some work remains to be done in this area since, even is a characterization of non-local minima is known, an efficient and stable algorithm might eschew the search completely given a simple test for existence.

2.5 Semidefinite relaxation

As we did for *TRS*, we derive the semidefinite relaxation of *2-TRS*. Details of the derivation are omitted when they are identical to the *TRS* case. We recall the primal problem,

$$\text{2-TRS} \qquad \min\left\{\mu(x) = x^t Q x + 2b^t x \mid x^t Q_1 x + 2b_1^t x - a_1 \leq 0, x^t x \leq \delta^2, x \in \mathbb{R}^n\right\}.$$

We homogenize by adding a new component to the x vector and move into the semidefinite cone by relaxing the rank one constraint to get

$$\textit{PSDP} \qquad \min\left\{\tilde{\mu}(Y) = \langle P, Y\rangle \;\middle|\; \langle P_1, Y\rangle \le a_1, \langle P_I, Y\rangle \le \delta^2, \langle E_{00}, Y\rangle = 1, Y \succeq 0\right\},$$

where

$$P = \begin{bmatrix} 0 & b^t \\ b & Q \end{bmatrix}, P_1 = \begin{bmatrix} 0 & b_1^t \\ b_1 & Q_1 \end{bmatrix}, P_I = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, Y = \begin{bmatrix} 1 & x^t \\ x & X \end{bmatrix}.$$

The matrix E_{00} has only one non-zero entry in the top left corner, and all matrices are in $\mathbb{R}^{(n+1)\times(n+1)}$. The corresponding dual program is

$$\textit{DSDP} \qquad \max\left\{\nu(\mu,\lambda) = -\mu - \lambda_1 a_1 - \lambda\delta^2 \;\middle|\; P + \lambda_1 P_1 + \lambda P_I + \mu E_{00} \succeq 0, \lambda \ge 0\right\},$$

a program we recognize as the Lagrangean dual, after homogenization, with an explicit semidefinite constraint. This leads to the following set of equalities, stated here explicitly for future reference.

Lemma 2.5.1 *The dual optimal values of 2-TRS denoted ν^*, and of DSDP, denoted $\tilde{\nu}^*$, are equal to each other and to the optimal value of the primal semidefinite program, $\tilde{\mu}^*$.*

Proof: The equality $\nu^* = \tilde{\nu}^*$ follows by construction of the semidefinite relaxation. The equality of $\tilde{\nu}^* = \tilde{\mu}^*$ follows from strong duality of a linear semidefinite pair under Slater's constraint qualification, which holds in this case. □

We recall the definitions of the feasible sets of interest and of the projector map used for *TRS* to simplify the next lemma relating the feasible set of the original problem to the set of the relaxation.

$$\begin{aligned} F &:= \{x \in \mathbb{R}^n, x^t x \le \delta^2, x^t Q_1 x + 2b_1^t x - a_1 \le 0\}, \\ \tilde{F} &:= \{Y \in \mathbb{S}_{n+1}, \langle E_{00}, Y\rangle = 1, \langle P_I, Y\rangle \le \delta^2, \langle P_1, Y\rangle \le a_1\}, \end{aligned}$$

$$P_R : \mathbb{S}_{n+1} \Rightarrow \mathbb{R}^n;\; P_R\left(\begin{bmatrix} a & x^t \\ x & X \end{bmatrix}\right) = x, \quad P_R^{-1} : \mathbb{R}^n \Rightarrow \mathbb{S}_{n+1};\; P_R^{-1}(x) = \begin{bmatrix} 1 & x^t \\ x & xx^t \end{bmatrix}.$$

Lemma 2.5.2 *The feasible set of 2-TRS, denoted F, and the mapping, under P_R, of the feasible set of PSDP, denoted $\tilde{F}$ are equal. In short, $F = P_R(\tilde{F})$.*

Proof: The proof follows from Lemma 1.4.2 applied to each constraint in turn. □

Solving the semidefinite relaxation of *2-TRS* therefore yields a feasible solution to the original problem. Moreover, in some cases, as we now see, this feasible solution is optimal.

Lemma 2.5.3 *If $Q \succeq 0$, then from Y, an optimal solution of PSDP, we can obtain $x = P_R(Y)$, an optimal vector for 2-TRS. Moreover, if the objective function is strictly convex, Y is rank one.*

Proof: The result follows from Lemma 1.4.3. □

Lemma 2.5.4 *If Y, the optimal solution of PSDP, is rank one, then $x = P_R(Y)$ solves 2-TRS.*

Proof: The result follows from Lemma 1.4.4. □

At this point, *TRS* and *2-TRS* begin to differ. In the former case we could, by moving in the nullspace of the Hessian of the Lagrangean, attain the optimal solution. This is not always possible in *2-TRS* as the following example shows. There can be an unsurmountable gap between the optimal values of *2-TRS* and of its Lagrangean (thus semidefinite) relaxation.

Example 2.5.5 An instance of *2-TRS* with a gap between its optimal value and the optimal value of the its semidefinite relaxation.

$$\mu^* = \min\left\{-10x_1^2 - 2x_2^2 + 12x_1 - 10 \mid x_1^2 + x_2^2 - 10x_1 - 4x_2 + 20 \leq 0, x_1^2 + x_2^2 - 36 \leq 0\right\}.$$

The optimal solution is $x^* = (6\ 0)^t$, with optimal value $\mu^* = -298$. The semidefinite relaxation, after some rearranging to simplify the presentation, is given by

$$\tilde{\mu} = \min\left\{\langle P_0, Y\rangle \mid \langle P_k, Y\rangle \leq 0, 1 \leq k \leq 2, \langle E_{00}, Y\rangle = 1, Y \succeq 0\right\},$$

$$P_0 = \begin{bmatrix} -10 & 6 & 0 \\ 6 & -10 & 0 \\ 0 & 0 & -2 \end{bmatrix}, P_1 = \begin{bmatrix} -36 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, P_2 = \begin{bmatrix} 20 & -5 & -2 \\ -5 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}, E_{00} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

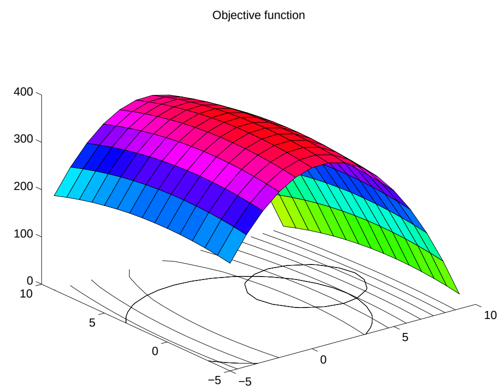


Figure 2.11: Primal objective function of Example 2.5.5.

From the optimal semidefinite primal solution, the first column yields $y = (5.48\ 0.3)^t$, after the projection discarding the homogenization component. The corresponding (strictly positive) optimal Lagrange multiplier vector is $\tilde{\lambda} = (1.2\ 8.8)^t$. Since Y is optimal for the SDP problem, it must be that the constraints of the relaxation are both active, i.e., $\langle P_1, Y\rangle = 0$, and $\langle P_2, Y\rangle = 0$. Yet the projection, given by the first column, satisfies no constraint with equality. Now we let

$$\bar{P} = \sum \lambda_i P_i = \begin{bmatrix} -292.8 & -6 & -2.4 \\ -6 & 10 & 0 \\ -2.4 & 0 & 10 \end{bmatrix}, \quad Z = \bar{P} + \mu E_{00} + P_0 = \begin{bmatrix} 0.64 & 0 & -2.4 \\ 0 & 0 & 0 \\ -2.4 & 0 & 8 \end{bmatrix},$$

and consider the Lagrangean.

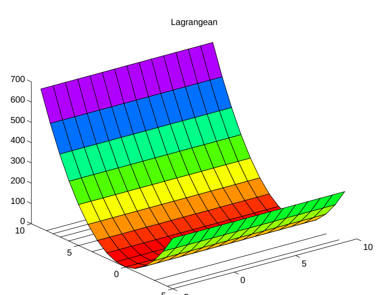


Figure 2.12: Lagrangean of Example 2.5.5 with optimal multipliers. Its Hessian is singular.

The Hessian of the Lagrangean is not full rank, so there is a direction in which we can move, as we did in the *TRS* case, while maintaining the same optimal objective value for the relaxation.

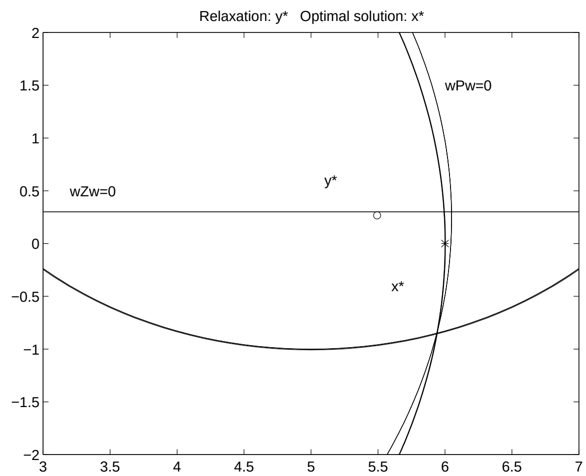


Figure 2.13: Corresponding to Example 2.5.5, the darker lines indicate the trust-regions. The optimal solution, x^*, is marked by a star, and the semidefinite relaxation, y^*, by a circle. The line $wZw = 0$ is the nullspace of the Hessian of the Lagrangean and $wPw = 0$ is the linear combination of the trust-regions given by the optimal multiplier vector λ^*.

We can move in the nullspace of the Lagrangean, the line $w^tZw = 0$, improving the objective value of *2-TRS*, until we hit the boundary of the feasible region, but we will not attain the optimal value. How close to the optimal solution the nullspace-restricted step moves is difficult to quantify. But, if nothing else, it produces a reasonable upper bound for the primal value.

2.6 Branch and bound for approximate solution

Whenever we are only concerned with a bound for the optimal value or an approximation to the optimal solution, the nullspace-restricted move described above is appropriate and we will use it while solving general nonlinear programs. But before we leave *2-TRS*, we give a few more results that potentially yield a better approximate solution and shed some light on a complete solution. This section should be viewed as tangent to the general direction of our work, and is stated here for completeness only.

We have hinted that a complete solution for *2-TRS* might involve looking at the global and non-global minimizers of each trust-region. We now describe how these points are related to the optimal solution we seek.

To simplify the discussion, we restate *2-TRS* in a slightly more general manner,

$$\textit{2-TRS} \qquad \min\Big\{q_0(x) \;\Big|\; q_1(x) \le 0, q_2(x) \le 0\Big\},$$

where q_1 and q_2 are convex quadratic constraints. We explicitly define the two implicit trust-region problems,

$$TRS_1 \qquad \min\Big\{q_0(x) \;\Big|\; q_1(x) \le 0\Big\} \qquad TRS_2 \qquad \min\Big\{q_0(x) \;\Big|\; q_2(x) \le 0\Big\}.$$

With this notation the next two results can be easily stated. They appeared, without proof and in a slightly different format, in Williamson thesis [73] and form the basis of her 2-dimensional, projected *CDT*-solving algorithm.

Lemma 2.6.1 *If a local minimum x of either trust-region problems TRS_1 or TRS_2 is feasible for 2-TRS, then x is a local minimum of 2-TRS. If, in addition, x is a global minimum of either trust-regions, then it is optimal for 2-TRS.*

Proof: Since x is a local minimum for one trust-region, there is no feasible direction improving the objective function within that trust-region. Reducing the feasible set can only reduce the set of feasible directions. Therefore, since x is feasible for *2-TRS*, it is a local minimum. If x is a global minimizer for one trust-region, by the same argument, it is global for *2-TRS*. □

Note that the converse is not necessarily true. A global minimizer of *2-TRS* need not be a global minimizer of either trust-regions. What the previous lemma implies is that looking for minimizers of the implicit trust-regions is, in those cases where such minima are feasible for *2-TRS*, sufficient to solve *2-TRS*. The other cases are handled by looking at the intersections.

Lemma 2.6.2 *If no local minima of either trust-regions TRS_1 or TRS_2 are feasible for 2-TRS, then the optimal solution x^* lies somewhere on the intersection of their boundaries.*

Proof: By way of contradiction, assume that the optimal solution for *2-TRS*, x, does not satisfy both constraints with equality and is not a local minimum of either trust-regions. We distinguish two cases.

In the first case, there are no active constraints and the objective is convex. Therefore x is the unconstrained minimizer. It must be optimal for both trust-regions, a contradiction.

In the other case, on the implicit trust-region problem of the single active constraint, since we assumed x was not optimal, there is a feasible direction improving the objective. Since the second constraint is inactive, the direction is also feasible with respect to *2-TRS*, contradicting the optimality of x. We conclude that x must lie on the intersection of the boundaries of both trust-regions. □

We could, in principle, solve *2-TRS* by looking at all global and non-global solutions of each trust-region and at all intersections of the constraints. But this is a daunting task if the problem is of any dimension higher than two. The set of local minima is finite but the intersection set need not be. Williamson did look at all such points because the problem she considered was two-dimensional and the number of intersections was at most four. An alternative was investigated by Heinkenschlos [28] who replaced the two constraints by a single constraint equivalent to the manifold of the intersections.

We contend ourselves with an approximate solution that we may find easily by looking at the global then the local minima, since we know there are a finite number of them, and then, by looking at one intersection point.

The general idea for the algorithm is to solve first the semidefinite relaxation and possibly apply the nullspace-restricted step. Whenever the objective is convex over the feasible set or when the optimal solution can be attained by the nullspace-restricted step, we are done. In all other cases we get a feasible point and bounds on the optimal value.

If the first step did not yield the optimal solution, it provided a feasible, strictly interior point. Starting from this point and using the bounds to stop the iterations whenever appropriate, we look for global minimizers of each trust-region. This generally involves few steps of the interior-point code. Either we find one such feasible global minimum and we are done, or we update our bounds.

If neither approach found the optimal solution, we look for non-global minima of each trust-region, again truncating the iterations with the help of the previously found bounds. If one non-global is feasible, we have a local minimum of *2-TRS* (that may fail to be optimal). If no local minima was found, we find the closest intersection to the strictly interior point found in step one. The usual method used to find such intersections is Gauss-Newton.

2.6.1 Generalized Newton step

The following is a slight digression on solving systems of equations, included here only because of an interesting relation between Gauss-Newton and a generalized Newton step. The intersection of two quadratic functions can be found by solving

$$q(x) = \begin{bmatrix} q_1(x) \\ q_2(x) \end{bmatrix} = 0,$$

where $q_i(x)$ are, say, the two quadratic constraints of *2-TRS*. The best-known approach to solve such a system is probably Gauss-Newton, applied to the 2-norm of q. We describe a somewhat more general approach and then show that Gauss-Newton is a only a special case of this approach.

We linearize q at $x^{(k)}$ and solve for a step from $x^{(k)}$ to $x^{(k+1)}$ by

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - [q'(x^{(k)})]^{-} q(x^{(k)}) \\ &= x^{(k)} - \begin{bmatrix} \nabla q_1(x^{(k)})^t \\ \nabla q_2(x^{(k)})^t \end{bmatrix}^{-} \begin{bmatrix} q_1(x^{(k)}) \\ q_2(x^{(k)}) \end{bmatrix}. \end{aligned}$$

This type of Newton step, based on a generalized inverse (indicated by $(\cdot)^{-}$) since $q'(x^{(k)})^t$ is a rectangular $(2 \times n)$-matrix, was investigated by Ben-Israel and Greville [4]. They show that any 2-inverse, under some additional boundedness conditions, will produce a convergent sequence $\{x^{(k)}\}$.

In the case where the gradients of q_1 and q_2 are linearly independent which, we recall, is not a requirement of the generalized Newton step, then $[q'(x^{(k)})]^-$ is full column rank and the limit point x^* of the sequence $\{x^{(k)}\}$ will therefore satisfy $q(x) = 0$.

An interesting question is the relation, if any, between this generalized-inverse Newton step and the classical Gauss-Newton step that we now derive.

First, as above, we linearize q at $x^{(k)}$ to get

$$\bar{q}(x) = q(x^{(k)}) + q'(x^{(k)})\delta,$$

for $\delta = x^{(k+1)} - x^{(k)}$ and we minimize the 2-norm of the linearization for the new point by

$$\begin{aligned} x^{(k+1)} &= \operatorname{argmin}\left\{||\bar{q}(x^{(k)})||^2\right\} \\ &= \operatorname{argmin}\left\{||q(x^{(k)})||^2 + 2[q'(x^{(k)})^t q(x^{(k)})]^t\delta + \delta^t q'(x^{(k)})^t q'(x^{(k)})\delta\right\}, \end{aligned}$$

a quadratic unconstrained program. Clearly the Hessian, $q'(x^{(k)})^t q'(x^{(k)})$, is positive semidefinite and if we assume that is is invertible (as we must in Gauss-Newton), we obtain the minimum by solving for stationarity,

$$q'(x^{(k)})^t q'(x^{(k)})\delta + q'(x^{(k)})^t q(x^{(k)}) = 0,$$

which leads to the step

$$x^{(k+1)} = x^{(k)} - \left[q'(x^{(k)})^t q'(x^{(k)})\right]^{-1} q'(x^{(k)})^t q(x^{(k)}).$$

The interesting fact arises when we compare the Gauss-Newton step to the previous, generalized-inverse Newton step. They are equal if and only if $\left[q'(x^{(k)})^t\right]^- = q'(x^{(k)})\left[q'(x^{(k)})^t q'(x^{(k)})\right]^{-1}$. It easy to see that this equation satisfies the four Penrose conditions. Let $A = q'(x^{(k)})^t$.

$$\begin{aligned} AA^-A &= AA^t[AA^t]^{-1}A &&= A \\ A^-AA^- &= A^t[AA^t]^{-1}AA^t[AA^t]^{-1} &&= A^- \\ (AA^-)^t &= (AA^t[AA^t]^{-1})^t &&= AA^- \\ (A^-A)^t &= (A^t[AA^t]^{-1}A)^t &&= A^-A \end{aligned}$$

Since there is only one matrix satisfying the Penrose equations, we conclude that the Gauss-Newton step is a special case of the generalized Newton step for the choice of the Moore-Penrose inverse. All of this comes together in the following pseudo-code.

ALGORITHM FOR BRANCH AND BOUND SOLUTION OF *2-TRS*

BB2-*TRS*$(Q, Q_1, b, b_1, a_1, \delta^2)$

Find feasible point $\tilde{x}$ and bounds by solving *SDP* relaxation

for each trust-region

$(x, \lambda) = TRS()$

if x is feasible

$x^* = x$

else

update the bound

fi

Solve by Martinez algorithm for non-global minimizer x

if x is feasible

update the bound and potential minimizer x^*

fi

endfor

if no optimal x^* was found

Solve by (generalized-inverse) Newton's method

$x^tQ_1x + b_1^tx - a_1 = 0; x^tx - \delta^2 = 0$ for x^*

fi

return(x^*)

This algorithm has the same overall structure but is nevertheless different from Williamson's [73] algorithm in the sense that it is applied to n-dimensional problems and uses bounds to truncate the searches for local minima. And, of course in that it does not guarantee an optimal solution, just an approximation.

2.7 A second step

One of the more interesting results of this chapter is the relation between the feasible set of the original problem (denoted F) and the set of feasible solutions to the semidefinite relaxation, after projection on the first column (denoted $\tilde{F}$). The result can be summarized as

$$\begin{aligned} F &\subseteq \tilde{F} \qquad \text{in all cases,} \\ F &= \tilde{F} \qquad \text{for convex } F. \end{aligned}$$

This relation allows us to develop an algorithm based on a primal-dual pair of semidefinite problems that will solve convex *2-TRS* programs and approximate non-convex instances. If the feasible region is convex, the algorithm always yields a feasible point, and both an approximate solution and bounds on the optimal value.

Each step of the algorithm solves a subproblem based on a semidefinite primal-dual pair. We describe, in a later chapter, one of many possible interior-point approaches used to solve such problems. This type of subproblem is the mainstay of the general nonlinear solver we describe in the next section.

The generalizations of *2-TRS* can be taken in different directions. We intend to consider multiple trust-regions, not in detail, but as the last stepping stone towards a general nonlinear solver.

Chapter 3

Fully quadratic programming

Sequential Quadratic Programming, denoted *SQP*, also known as Recursive Quadratic Programming, falls under the heading of Lagrange [38] or Newton-Lagrange [21] methods and is arguably the most efficient general-purpose algorithm for medium size nonlinear constrained programs [66], [8]. With solid theoretical foundations where, with the appropriate quadratic subproblem, the method can be viewed as an extension of Newton or quasi-Newton algorithms to constrained optimization, it is also very successful in the practical, even the commercial world. In a recent [46] list of over thirty optimization packages, some variation of *SQP* appears prominently as the basic algorithm.

Yet the very existence of these many variations indicates that the last word on *SQP* has not been written. Recent research has produced variations known as SL_1QP [11], [21] for their choice of non-differentiable merit function and FSQP [48] for a method where iterates are kept within the feasible region. And much of the current research aims to apply the method to large-scale problems [25].

The approximation of the objective function to second order, and of the constraints only up to first order, has been viewed as an incoherence of the method. But the subproblem of a quadratic objective function subjected to quadratic constraints has long been considered intractable, so that the attempts to exploit the curvature of the constraints were, if efficient, somewhat complex to implement [11] [12].

We investigate how semidefinite programming can be used to produce and to solve a second-order subproblem within the general framework of an *SQP* algorithm. Most of the results concern the convex case but some extensions to non-convex programs with convex feasible sets will be given. This is work in progress towards a general nonlinear program solver.

3.1 Traditional sequential programming

In order to set the notation used throughout and to motivate the work, we start by describing the standard *SQP* approach. The original algorithm dates from Wilson's [74] dissertation in 1963 but was made well-known by Beale [3] a few years later.

Consider the general, nonlinear programs with equality and inequality constraints

$$\textit{NEP} \qquad \min\left\{f(x) \mid h(x)=0, x\in\mathbb{R}^n\right\} \qquad \text{and} \quad \textit{NLP} \qquad \min\left\{f(x) \mid g(x)\le 0, x\in\mathbb{R}^n\right\},$$

where $f:\mathbb{R}^n \Rightarrow \mathbb{R}$, and $h,g:\mathbb{R}^n \Rightarrow \mathbb{R}^m$. We sometimes write vector-valued functions, like $h(x)$, as

$$h(x) = (h_1(x)\ h_2(x)\ \dots\ h_m(x))^t.$$

We define the Lagrangean of *NEP* as $\mathcal{L}(x,\lambda) := f(x) + \lambda^t h(x)$. The first-order necessary conditions for *NEP* at an optimal point x^* state $\nabla_x \mathcal{L}(x^*,\lambda) = 0$. Together with feasibility (equivalent to $\nabla_\lambda \mathcal{L}(x^*,\lambda) = 0$), stationarity expands to

$$\begin{aligned} \nabla f(x^*) + h'(x^*)^t\lambda &= 0, \\ h(x^*) &= 0, \end{aligned}$$

where $\lambda = (\lambda_1\ \lambda_2\ \dots\ \lambda_m)^t$ is the vector of (free) Lagrange multipliers. To simplify the exposition, we use $h'(x)$ to denote $[\nabla h_1(x) \nabla h_2(x) \dots \nabla h_m(x)]^t$, the Jacobian of h.

An iterative attempt at the non-linear system above by Newton's method produces

$$\begin{bmatrix} \nabla^2 f(x^{(k)}) + \sum \lambda_i^{(k)} \nabla^2 h_i(x^{(k)}) & h'(x^{(k)}) \\ h'(x^{(k)})^t & 0 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(x^{(k)}) - h'(x^{(k)})^t \lambda^{(k)} \\ -h(x^{(k)}) \end{bmatrix},$$

where $\delta_x = x^{(k+1)} - x^{(k)}$ and $\delta_\lambda = \lambda^{(k+1)} - \lambda^{(k)}$. The usual simplification, at this point, is to let $\lambda^{(k+1)} = \lambda^{(k)} + \delta_\lambda$ and $d = \delta_x$, to obtain what we will refer to as the First-Order Newton Step,

$$\textit{(FONS)} \qquad \begin{bmatrix} \nabla^2 \mathcal{L}(x^{(k)}, \lambda^{(k)}) & h'(x^{(k)}) \\ h'(x^{(k)})^t & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} -\nabla f(x^{(k)}) \\ -h(x^{(k)}) \end{bmatrix}.$$

This system produces a direction d and a new vector of Lagrange multipliers $\lambda^{(k+1)}$.

An important remark is that the system of equations *(FONS)* can also be derived as the first-order necessary conditions of the quadratic program

$$\begin{array}{lll} QP \quad \min & q(d) = & f(x^{(k)}) + \nabla f(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 \mathcal{L}(x^{(k)}, \lambda^{(k)}) d \\ \qquad \text{s.t.} & l_i(d) = & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^t d = 0, \qquad 1 \le i \le m, \end{array}$$

hereafter known as the QP subproblem. Stationarity of the Lagrangean of QP yields the first line of *(FONS)*, and feasibility yields the second line. This is why SQP is viewed as an extension of Newton's method to constrained optimization.

The expression QP, for *quadratic programming*, is somewhat unfortunate since there is very little worthy of the term quadratic in the above system. There are, in general, many more *linear* constraints than the one and only quadratic function we find in the objective. A better description would have been *almost-linear programming*. In fact, the very efficient active set approaches [7] used to solve QP can be viewed as generalizations of the simplex method. They are therefore much closer in spirit to linear than to nonlinear programming. But we shall abide by tradition. A quadratically constrained quadratic program, examples of which we will shortly see, will therefore be called a *fully* quadratic program and will be denoted by Q^2P.

There are advantages gained by solving the QP subproblem by some optimization code converging to a solution of the first-order condition of *NEP*, as opposed to solving the system of equations *(FONS)* by a linear solver: The iterates of the optimization subproblem, under the assumption that the second-order sufficient condition holds for the original problem, will converge to a minimum of the objective function while solving the system guarantees only stationarity. The subproblem eliminates undesirable stationary points. Moreover, the *QP* program, described above for *NEP*, generalizes easily to *NLP*, the inequality constrained program, while *(FONS)* does not.

We must recognize some characteristics of the *QP* subproblem. A Taylor first-order approximation of the constraint defines the feasible set while a second-order expansion of the objective, to which we add second-order terms of the constraints, completes the problem definition. These second-order terms are essential. A linear objective in the original problem, for example, may fail to have a solution if constrained only by linear approximations, as the following example illustrates.

Example 3.1.1 *The second-order terms are essentials in the program (from Fletcher [21]),*

$$\min\left\{-x_1 - x_2 \;\middle|\; 1 - x_1^2 - x_2^2 = 0\right\},$$

since a linear-linear subproblem, from point $(-1 \;\; -2)^t$, *for direction* $(d_1 \; d_2)^t$ *would yield*

$$\min\left\{-d_1 - d_2 \;\middle|\; -4 + 2d_1 + 4d_2 = 0\right\},$$

a badly-defined problem. Some curvature information is necessary.

We can forgo discussion of the line search under the assumption that a full step is taken at each iteration. But this is justified only if the initial estimate x is close enough to the optimal solution x^*. In general, the *SQP* approach relies on a merit function $\varphi(x, \lambda)$, reduced at each iteration and minimized when the system of first-order conditions *(FONS)* is satisfied. This function, ideally, has only global minima and is exemplified, for convex *NEP* programs, by

$$\varphi(x, \lambda) = \frac{1}{2}||\nabla f(x) + \sum_i \lambda_i \nabla h_i(x)||^2 + \frac{1}{2}||h(x)||^2.$$

Clearly, if an algorithm decreases this function to zero, it must have found a feasible solution satisfying the first-order conditions of *NEP*. In general, a well-behaved merit function has a local minimum where the constrained problem has a solution and it must allow the line search to accept a full step, at least asymptotically.

This line search procedure is expressed in the following algorithms as

$$\alpha = \text{linesearch}(\varphi(x^{(k)}, \lambda^{(k)}), d).$$

This is meant to suggest that the procedure minimizes, perhaps approximately, the merit function φ, from the current iterate $(x^{(k)}, \lambda^{(k)})$, in the direction d, and returns the step length α corresponding to this one-dimensional minimization. In theoretical works, line searches return either the exact minimum or some other length guaranteed to satisfy the Goldstein-Armijo or Powell conditions, yet, in practical algorithms, they are often relaxed.

Sequential Quadratic Programming Algorithm

SQP$(f, \nabla f, \nabla^2 f, h, \nabla h_i, \nabla^2 h_i, x^{(0)}, \lambda^{(0)})$

  do

    $d \in \text{argmin}\{\nabla f(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 f(x^{(k)}) d : h_i(x^{(k)}) + \nabla h_i(x^{(k)})^t d = 0, 1 \leq i \leq m, d \in \mathbb{R}^n\}$

    $\alpha = \text{linesearch}(\varphi(x^{(k)}, \lambda^{(k)}), d)$

    $x^{(k+1)} = x^{(k)} + \alpha d$

    $k = k + 1$

    Estimate new Lagrange multipliers

  until convergence

return$(x^{(k)}, \lambda^{(k)})$

We leave undefined the convergence criterion of this algorithm since practical considerations cloud the issue. In theory, since *SQP* converges to a point satisfying the first-order necessary condition of *NEP*, it is appropriate to use some tolerance $\epsilon > 0$ and define convergence as

$$||\nabla f(x^{(k)}) + \sum_i \lambda_i \nabla h_i(x^{(k)})|| + ||h(x^{(k)})|| \leq \epsilon,$$

where ϵ is appropriately scaled. A simpler test on the step length might be more practical, less costly and just as appropriate; while a more complex test, involving the merit function, might have better theoretical justification. (For a wealth of details about termination criteria, see Gill, Murray and Wright [50], §8.2.3.)

It is instructive to look at a few iterations of a convex program solved by *SQP*. Consider
Example 3.1.2

$$\min\left\{-x_1 - x_2 \mid x_1^2 + x_2^2 - 1 \leq 0, x_1^2 + x_2^2 \leq 0\right\},$$

with optimal solution $(1/\sqrt{2}, 1/\sqrt{2})^t$. Starting from initial point $(1/2, 1)^t$, close enough to the optimal solution, a naive *SQP* approach, without any line search, solved this program to five significant digits in four iterations. The algorithm stopped when the *QP* subproblem could find no direction for improvement, implying that the current x satisfied the first-order conditions.

Iter	$(x_1, x_2)^t$	$(\lambda_1, \lambda_2)^t$	$(d_1, d_2)^t$
1	(.5 1)	(+0.0000 +0.0000)	(+0.4167 -0.3333)
2	(+0.9167 +0.6667)	(+0.3333 +0.6667)	(-0.1695 +0.0196)
3	(+0.7471 +0.6863)	(+0.0000 +0.7304)	(-0.0384 +0.0205)
4	(+0.7088 +0.7068)	(+0.0000 +0.7067)	(-0.0017 +0.0003)
5	(+0.7071 +0.7071)	(+0.0000 +0.7071)	(+0.0000 +0.0000)

Figure 3.1 illustrates the iterations of *SQP*. The curves bound the original feasible set and the lines represent the linear constraints of the *QP* subproblem. The circles ($\circ$) are the successive iterates converging to the optimal solution ($*$). And the arrows represent the solution of the *QP* subproblem. We note that the feasible region of the linear approximation of the *QP* subproblem always contains the original feasible region. This is true in general of convex programs.

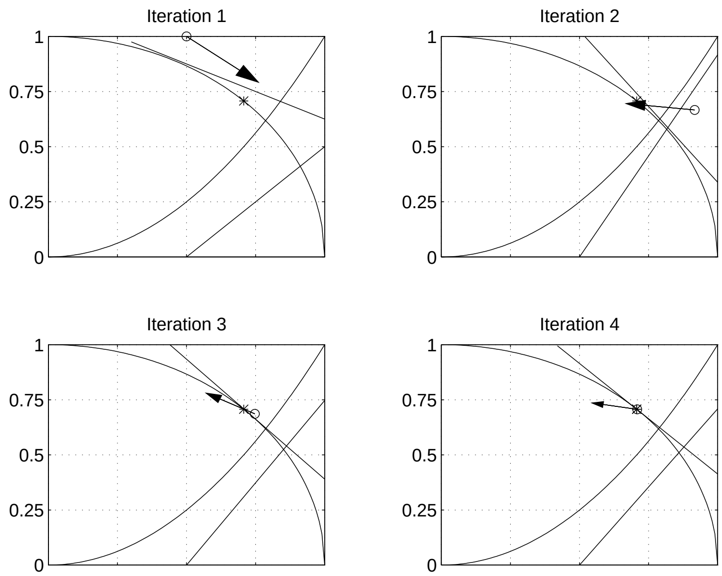


Figure 3.1: Iterations of *SQP* on Example 3.1.2 from initial point $(\frac{1}{2}\ 1)^t$ to optimal solution $(0.7071\ 0.7071)^t$, with unit step-lengths at every iteration.

Lemma 3.1.3 *If the feasible set of a nonlinear program, $F := \{x \mid g(x) \leq 0\}$, described by the convex function $g : \mathbb{R}^n \Rightarrow \mathbb{R}^m$, is approximated at the point x_0 by the first-order Taylor expansion of g, as in*

$$\bar{F} := \{x \mid g_i(x_0) + \nabla g_i(x_0)^t(x - x_0) \leq 0,\ 1 \leq i \leq m\},$$

then $F \subseteq \bar{F}$.

Proof: For any $x \in F$, the first-order characterization of convexity, namely

$$\nabla g_i(x_0)^t(x - x_0) \leq g_i(x) - g_i(x_0),\ 1 \leq i \leq m,$$

implies that we have $g_i(x_0) + \nabla g_i(x_0)^t(x - x_0) \leq g_i(x) \leq 0$, for all $1 \leq i \leq m$. We conclude that $x \in \bar{F}$ and, therefore, $F \subseteq \bar{F}$. □

There are a few problems with this algorithm, most of them due to the fact that the linear approximations are not good unless we are already very close to the optimal solution. Consider, for example starting the iterations, on the previous example, from a point close to the left-hand side intersection of the constraints. The resulting *QP* subproblem is then unbounded. Some modification, adding a trust region for example, is required, making practical implementations of *SQP* much more complex than our naive description would suggest.

But there is more. The Lagrange multipliers are not, in most implementations of *SQP*, a by-product of the subproblem. Although the multipliers resulting from *QP* can be added to the previous estimates used in the objective function, most authors suggest solving a separate least square problem [50] to get a better approximation of the "true" Lagrange multipliers.

We will return to these considerations when we describe the subproblem we wish to investigate since our intention is not to survey all the possible variations required to make *SQP* work in practice but rather to indicate how a different subproblem does away with some of the difficulties (unbounded problem, inaccurate feasible set, poor multipliers) while converging faster.

3.2 Higher-order expansions

We have described some of the problems associated with the quadratic program used in traditional *SQP*. After these negative considerations, to motivate positively the replacement subproblem that forms the basis of our recursive algorithm, we now consider Taylor expansions in relation with Newton's method.

We recall that the *QP* subproblem is used in the *SQP* approach because it implicitly produces a Newton step for a system of equations describing the first-order conditions. Newton's method for solving $h(x) = 0$ can be viewed as an expansion of the function h to first order around $x^{(k)}$, i.e.,

$$h(x) = h(x^{(k)}) + h'(x^{(k)})\delta + o(||\delta||),$$

which is then truncated to produce the linear system

$$h(x^{(k)}) + h'(x^{(k)})\delta = 0,$$

solved for δ. The next iterate is then obtained by $x^{(k+1)} = x^{(k)} + \delta$, and the process is repeated until convergence.

Some iterative methods using higher-order derivatives have been known for a long time. Interesting examples can be found in a 1870 paper by Schröder, translated from the German by Stewart[65]. In the one-dimensional case, the simplest extension of Newton's method is a second-order expansion of h,

$$h(x^{(k)}) + h'(x^{(k)})\delta + \frac{1}{2}h''(x^{(k)})\delta^2 = 0,$$

which is then solved to yield

$$\delta = \frac{-h(x^{(k)}) \pm \sqrt{[h'(x^{(k)})]^2 - 2h(x^{(k)}) \cdot h''(x^{(k)})}}{h''(x^{(k)})},$$

by the quadratic formula. The two possible solutions for δ are then investigated separately. This approach converges faster than the first-order Newton's method (assuming that the second derivative does not vanish). If we forget for a moment that there is no simple quadratic formula to solve for δ in the case where it lies in a space of dimension greater than one, we can nevertheless show in a very general fashion this faster convergence.

Say that x^* is a root of $h(x) = 0$. Let the error at step k be $d^{(k)} = x^* - x^{(k)}$ and let $p_i^r(d^{(k)})$ be a truncated Taylor expansion, around $x^{(k)}$, of the function $h_i(x)$. For example, if $r = 2$, the case that concerns us in most of this work, the expansion is

$$p_i(d^{(k)}) = h_i(x^{(k)}) + \nabla h_i(x^{(k)})^t d^{(k)} + \frac{1}{2}{d^{(k)}}^t \nabla^2 h_i(x^{(k)}) d^{(k)} + o(||d^{(k)}||^2).$$

With an explicit error term, we can therefore write

$$0 = h(x^*) = \begin{bmatrix} h_1(x^*) \\ \cdots \\ h_m(x^*) \end{bmatrix} = \begin{bmatrix} p_1^r(d^{(k)}) \\ \cdots \\ p_m^r(d^{(k)}) \end{bmatrix} + \begin{bmatrix} e_1^r(x^{(k)}, d^{(k)}) \\ \cdots \\ e_m^r(x^{(k)}, d^{(k)}) \end{bmatrix} ||d^{(k)}||^r,$$

where $e(x^{(k)}, d^{(k)})$ is a vector of error terms that satisfy $e_i(x^{(k)}, d^{(k)}) \to 0$ as $||d^{(k)}|| \to 0$. If we assume that $m < n$ and that we can solve a truncated system, then a "Newton" step of order r is given by the solution of

$$P^r(\delta^{(k)}) := \begin{bmatrix} p_1^r(\delta^{(k)}) \\ \dots \\ p_m^r(\delta^{(k)}) \end{bmatrix} = 0.$$

Subtraction of the two equations above yields

$$\begin{bmatrix} p_1^r(\delta^{(k)}) \\ \dots \\ p_m^r(\delta^{(k)}) \end{bmatrix} - \begin{bmatrix} p_1^r(d^{(k)}) \\ \dots \\ p_m^r(d^{(k)}) \end{bmatrix} = \begin{bmatrix} e_1^r(x^{(k)}, d^{(k)}) \\ \dots \\ e_m^r(x^{(k)}, d^{(k)}) \end{bmatrix} ||d^{(k)}||^r.$$

We now apply the Mean Value Theorem to each p_i to yield

$$\begin{bmatrix} e_1^r(x^{(k)}, d^{(k)}) \\ \dots \\ e_m^r(x^{(k)}, d^{(k)}) \end{bmatrix} ||d^{(k)}||^r = \begin{bmatrix} (p_1^r)'(\xi_1) \\ \dots \\ (p_m^r)'(\xi_m) \end{bmatrix} (\delta^{(k)} - d^{(k)}),$$

where each $\xi_i \in [\delta^{(k)}, d^{(k)}]$.

To simplify the notation, we introduce the symbol P' for the $m \times n$ matrix,

$$P' := \begin{bmatrix} (p_1^r)'(\xi_1) \\ \dots \\ (p_m^r)'(\xi_m) \end{bmatrix}.$$

If we assume that σ_1, the smallest singular value of P', satisfies $\sigma_1 \geq \epsilon > 0$, for all values of $\xi_i \in [\delta^{(k)}, d^{(k)}]$, and that we started sufficiently close to the root of $h(x) = 0$, then a Moore-Penrose inverse exist and has a spectral norm bounded away from infinity [1]. We will denote this inverse by $[P']^\dagger$.

This allows us to quantify the decrease of the distance of $x^{(k)}$ to x^* for a "Newton" method of

[1] If $r = 1$, for example, this requires the gradients of the h_i to be linearly independent at x^*, a familiar constraint qualification.

order r. The rate of decrease is larger whenever r is larger; that is, whenever h is approximated by a Taylor polynomial of higher-order. This conclusion follows from

$$h^{(k+1)} = x^* - x^{(k+1)} = x^* - x^{(k)} - \delta^{(k)} = \delta^{(k)} - d^{(k)} = [P']^\dagger e(x^{(k)}, d^{(k)}) ||d^{(k)}||^r,$$

so that, by considering compatible norms (Euclidean vector norm and spectral matrix norm), we get

$$||\delta^{(k)} - d^{(k)}|| = ||[P']^\dagger e(x^{(k)}, d^{(k)})|| \; ||d^{(k)}||^r \leq ||[P']^\dagger|| \; ||e(x^{(k)}, d^{(k)})|| \; ||d^{(k)}||^r.$$

As we stated above, for each row of the error vector, $e_i(x^{(k)}, d^{(k)}) \to 0$ as $||d^{(k)}|| \to 0$. The second factor on the right-hand side of the inequality is therefore bounded. And, under the assumptions made for P', the inverse satisfies $||[P']^\dagger|| \leq K < \infty$, for some scalar K. We therefore have a q-convergence rate of r.

For future reference, we restate without further proof the last expression for the rate of convergence of a Newton-type method of order r.

Proposition 3.2.1 *Say that h is an analytic function with root x^*, approximated at $x^{(k)}$ by a Taylor polynomial of order r, denoted by $P^r(d^{(k)})$ with error $d^{(k)} = x^* - x^{(k)}$. Under the assumptions described above there is a neighborhood where the solutions $\delta^{(k)}$ of $P^r(\delta^{(k)}) = 0$ exist and form a convergent sequence with q-convergence rate r.*

Traditional sequential quadratic programming solves iteratively a system of equations describing stationarity of the Lagrangean. Each iteration updates not only x, the primal solution, but also λ the Lagrange multipliers. From a second-order approximation of the constraints, we expect, possibly better primal iterates, but certainly better multipliers. Such an approximation applies more equal weight to primal and dual variables, in contrast to *SQP*, an algorithm biased towards primal variables. The next section formalizes the expected improvement. But informally, if an analytic function can be replaced by its infinite Taylor expansion, then the higher the degree of the polynomial approximation, the better the estimates.

3.3 Quadratic approximations of nonlinear programs

Recall that the standard *SQP* subproblem approximated the objective function to second order yet approximated the constraints only to first order. Some attempt is made to include curvature information in the objective function but this is done using the previous Lagrange multipliers.

We wish a better balanced, yet tractable, subproblem where the feasible region is also a second-order approximation. As the original subproblem considered was called the *QP* subproblem, we will call this program the Q^2P subproblem. Such a subproblem has often been considered before, but has, just as often, been discarded as unsolvable. One notable exception is an algorithm by Maany [40] developed, interestingly enough, because the standard *SQP* approach failed on the highly nonlinear orbital trajectory problems they were studying. (See Dixon, Hersom and Maany [16].)

Before we attempt the Q^2P subproblem, we will precisely construct it and analyze the properties it possesses that make it an attractive approximation to a nonlinear program.

3.3.1 Feasible region

First we investigate the feasible region of our subproblem. Consider a vector $x^{(k)} \in \mathbb{R}^n$, an estimate of the primal solution. Expand the functions of *NLP* by second-order Taylor polynomials and express

$$\begin{array}{llrcll} \textit{NLP-}Q^2P & \min & q_0(d) & = & \nabla f(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 f(x^{(k)}) d & \\ & \text{s.t.} & q_j(d) & = & g_j(x^{(k)}) + \nabla g_j(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 g_j(x^{(k)}) d \leq 0, & 1 \leq j \leq m. \end{array}$$

Since the subproblem above differs from the traditional *QP* subproblem mostly in the feasible region it describes, we need to investigate this region in some detail. First, we note the absence of any sort of active set strategy. All the constraints of *NLP* are included in *NLP-*Q^2P. This simplifies not only the notation, but more importantly, the results about optimality conditions.

Lemma 3.3.1 *If a convex program has a feasible region $F := \{x|g(x) \leq 0\}$, then the feasible region of the quadratic approximation given by*

$$\hat{F} := \{x|g_i(x_0) + \nabla g_i(x_0)^t(x - x_0) + \frac{1}{2}(x - x_0)^t \nabla^2 g_i(x_0)(x - x_0) \leq 0,\ 1 \leq i \leq m\},$$

is contained within the feasible region of the linear approximation

$$\bar{F} := \{x|g_i(x_0) + \nabla g_i(x_0)^t(x - x_0) \leq 0,\ 1 \leq i \leq m\}.$$

Proof: Since $\bar{F}$ is a linear approximation to F, it is also a linear approximation to $\hat{F}$. The result therefore follows by Lemma 3.1.3. □

It would be fortunate if the quadratic approximations were always between the original feasible regions and the linear approximations. Unfortunately, this is not the case, even for convex programs. Consider the epigraph of $y = e^x$ approximated at $x = 1.5$ as in the following figure.

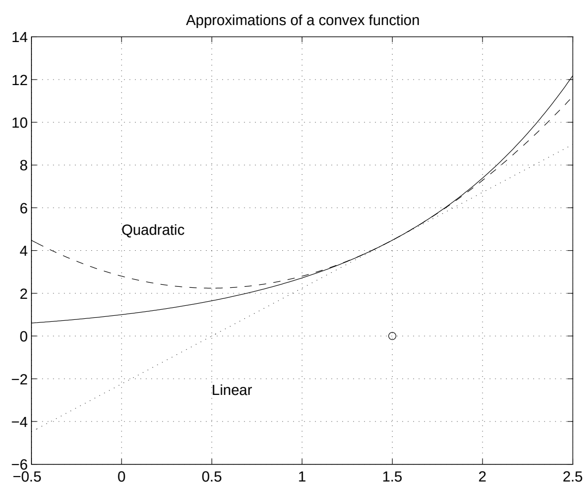


Figure 3.2: Linear and quadratic approximations of the convex function $y = e^x$ at $x = 1.5$.

Even if the feasible region of the quadratic approximation does not always include the original feasible region, it is closer in some sense to that region, for the Taylor residual is smaller. Also, since the second-order feasible region is within the linearly enclosed region, a bounded QP subproblem implies a bounded second-order subproblem. But note that the reverse is false so that Q^2P may be bounded while QP is not.

As as aside, there is a sense in which the semidefinite relaxation yields a feasible set somewhere in between the linear and the quadratic approximations. It is true, as we have seen in Lemma 3.4.2, that the first column of the semidefinite feasible solutions is isomorphic to the feasible set of the quadratically constrained program. But, as the matrices need not be rank one, there are more feasible solutions in the lifted space. The relation between the possible rank of the optimal solutions and the number of constraints was investigated by Pataki [49], but we will not pursue it further here.

3.3.2 Second-order Lagrange multiplier estimates

In traditional *SQP*, the multipliers are not usually by-products of the iterations. Yet, since they are essential in the formulation of the objective function, they must be reasonably accurate.

We recall that a pair of vectors x^* and λ^*, optimal for *NLP*, are related by the stationarity equation,

$$\nabla f(x^*) + \sum \lambda_i \nabla g_i(x^*) = 0.$$

This condition suggests that the optimal solution λ of the least-square problem,

$$\min\left\{ \|\nabla f(x^{(k)}) + \sum \lambda_i \nabla g_i(x^{(k)})\|_2^2 \mid \lambda \in \mathbb{R}^m \right\},$$

might provide an appropriate estimate of the true multiplier. An estimate which improves as $x^{(k)}$ approaches feasibility and the right active set is identified. This is the path taken by most implementations of *SQP*.

In a section of their book devoted to the identification of accurate multipliers, Gill, Murray and Wright [50] pursue this further and suggest aiming for second-order multiplier estimates: The approach is to let $d = x^* - x^{(k)}$ and expand the stationarity condition of *NLP*, around $x^{(k)}$, by a

Taylor polynomial of first order to get

$$\nabla f(x^{(k)}) + \nabla^2 f(x^{(k)})d + \sum \lambda_i(\nabla g_i(x^{(k)}) + \nabla^2 g_i(x^{(k)})d) + o(\|d\|^2) = 0,$$

or, using the Lagrangean,

$$\nabla f(x^{(k)}) + \nabla^2 \mathcal{L}(x^{(k)}, \lambda^*)d + \sum \lambda_i^* \nabla g_i(x^{(k)}) + o(\|d\|^2) = 0.$$

Gill, Murray and Wright note at this point that it is impossible to estimate λ^* directly from the above equation for two reasons: First, d is unknown; second, components of λ^* are buried inside the Hessian of the Lagrangean. They reason that the best available multipliers λ and an approximating step d used in a least-square problem such as

$$\min\left\{\|\nabla f(x^{(k)}) + \nabla^2 \mathcal{L}(x^{(k)}, \lambda)d + \sum \eta_i \nabla g_i(x^{(k)})\|_2^2 \mid \eta \in \mathbb{R}^m\right\},$$

would provide a vector η, deemed a second-order estimate of λ^* if d is sufficiently small and λ is, at least, a first-order estimate of λ^*.

This is where the Q^2P subproblem yields another advantage over *QP*. From stationarity of *NLP*-Q^2P, at optimal vectors d and λ, we obtain

$$\nabla f(x^{(k)}) + \nabla^2 \mathcal{L}(x^{(k)}, \lambda)d + \sum \lambda_i \nabla g_i(x^{(k)}) = 0.$$

These optimal multipliers λ therefore solve the second-order least-square problem for the given d. One of the two concerns of Gill, Murray and Wright, namely that the correct multipliers are buried in the Hessian of the Lagrangean is implicitly taken care of. We need only to assume that $x^{(k)}$ is close to x^* to conclude that the multipliers obtained from the Q^2P subproblem are second-order estimates of the true optimal multipliers. Without solving an additional least-square problem, Q^2P yields valuable dual variables in tandem with primal updates.

3.3.3 Primal step

We now turn our attention to the the vector d obtained from Q^2P, to qualify its value as a primal update. In this section we assume that the strong constraint qualification holds for *NLP*: at optimality, the gradients of the active constraints are linearly independent.

Lemma 3.3.2 *Assume that $x^{(k)}$ is feasible for NLP. If the NLP-Q^2P subproblem is solved by $d = 0$ with multipliers λ, then the pair of vectors $x^{(k)}$ and λ satisfies the first-order conditions of NLP.*

Proof: Since the components of the multipliers λ satisfy the necessary conditions of *NLP-Q^2P*, they are nonnegative. By complementarity for *NLP-Q^2P*,

$$\lambda_i \left(g_i(x^{(k)}) + \nabla g_i(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 g_i(x^{(k)}) d \right) = 0, \qquad 1 \le i \le m,$$

and by $d = 0$, this reduces to

$$\lambda_i g_i(x^{(k)}) = 0, \qquad 1 \le i \le m,$$

complementarity for *NLP*.

Finally, stationarity of the Lagrangean of *NLP-Q^2P* implies

$$\nabla f(x^{(k)}) + \nabla^2 f(x^{(k)}) d + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^{(k)}) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x^{(k)}) d = 0.$$

Again by $d = 0$, this reduces to stationarity of *NLP*

$$\nabla f(x^{(k)}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^{(k)}) = 0.$$

Therefore all first-order conditions of *NLP* are satisfied. □

This shows that the Q^2P subproblem does at least as well as the QP subproblem since they both solve the first-order conditions. In fact, Q^2P does better.

Lemma 3.3.3 *Assume that $x^{(k)}$ is feasible for NLP. If the NLP-Q^2P subproblem is solved by $d = 0$ with multipliers λ, then the pair of vectors $x^{(k)}$ and λ satisfies the second-order conditions of NLP.*

Proof: The necessary second-order conditions of *NLP-Q^2P* imply that the matrix

$$\nabla^2 f(x^{(k)}) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x^{(k)})$$

is positive semidefinite on the subspace tangent to the active constraints, namely the elements $i \in \{1, \ldots, m\}$ for which

$$g_i(x^{(k)}) + \nabla g_i(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 g_i(x^{(k)}) d = 0.$$

But, as $d = 0$ this reduces to $g_i(x^{(k)}) = 0$, the set of constraints of *NLP* active at $x^{(k)}$. □

The above two lemmata imply that if $d = 0$ solves *NLP-Q^2P*, we have solved for a pair of primal-dual vectors satisfying first and second-order conditions of *NLP*. The reverse also holds.

Lemma 3.3.4 *Assume that $x^{(k)}$ and λ satisfy the first and second-order necessary conditions of NLP. Then the pair of vectors $d = 0$, λ satisfy the first and second-order conditions of NLP-Q^2P.*

Proof: First we note that $d = 0$ is feasible for *NLP-Q^2P* since $x^{(k)}$ is feasible for *NLP*. Complementarity follows similarly. For all $1 \leq i \leq m$,

$$\lambda_i \left(g_i(x^{(k)}) + \nabla g_i(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 g_i(x^{(k)}) d \right) = \lambda_i g_i(x^{(k)}) = 0,$$

where the first equality follows from the choice $d = 0$ and the last, from complementarity for *NLP*. Stationarity for *NLP-Q^2P* reads

$$\left(\nabla^2 f(x^{(k)}) + \sum \lambda_i \nabla^2 g_i(x^{(k)}) \right) d + \nabla f(x^{(k)}) + \sum \lambda_i \nabla g_i(x^{(k)}) = \nabla f(x^{(k)}) + \sum \lambda_i \nabla g_i(x^{(k)}) = 0,$$

where the last equality is stationarity of *NLP*. Finally, the second-order condition holds since the Hessians of the Lagrangean of both problems are identical. □

We can summarize lemmata 3.3.4, 3.3.3 and 3.3.2: Under the usual constraint qualification of linear independence of the gradients, the pair of vectors $x^{(k)}, \lambda$ satisfy first and second-order necessary conditions of *NLP* if and only if the pair $d = 0, \lambda$ satisfy first and second-order conditions of *NLP-Q^2P*. We emphasize here that there is no question yet of solving this *NLP-Q^2P* subproblem, by semidefinite relaxation or otherwise. We are simply justifying it's existence, i.e., showing why it is a better subproblem than the standard *QP* subproblem.

As an aside, we can also describe in positive terms what a non-zero solution d of *NLP-Q^2P* yields from a feasible point $x^{(k)}$ of a convex program. We will not pursue this much as we do not intend to force feasibility of the iterates (in the manner of Panier and Tits [48]). But it does suggest that an interior-point type algorithm for convex programs is possible in the framework we describe. It might be interesting to implement this idea and compare its practical behavior to other algorithms like, for example, Vanderbei's LOCO, the restriction of LOQO [70] to convex optimization problems. Given convex constraints of high curvature, an interior-point *SQ^2P* might perform well.

Lemma 3.3.5 *Suppose that q_0, the approximation of f at $x^{(k)}$, is convex and that $q_0(d) \neq 0$ solves NLP-Q^2P, then d is a descent direction for f.*

Proof: Since $d = 0$ is a feasible vector for *NLP-Q^2P* with objective value 0, we conclude that the optimal value is negative. Therefore,

$$\nabla f(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 f(x^{(k)}) d < 0$$

or,

$$\nabla f(x^{(k)})^t d < -\frac{1}{2} d^t \nabla^2 f(x^{(k)}) d \leq 0,$$

where the last inequality follows from the convexity of q_0. □

Whether we can always find such improving directions is related to global convergence of the algorithm, an issue we will leave open while we attack the practicalities of solving *NLP-Q^2P*.

At this point we have shown some of the characteristics of the *NLP-Q^2P* subproblem. It may be worthwhile to repeat that an algorithm iterating exclusively on feasible points is possible. But one strength of *SQP*, in most of it's variations, is not to require feasibility until convergence. We wish to consider iterations of our algorithm based on infeasible points. We therefore address now the more complex problem of solving *NLP-*Q^2P from arbitrary starting points.

3.4 Semidefinite relaxation of multiple trust-regions

The solution method of *NLP-*Q^2P we suggest in this section is a generalization of the previous chapter concerning the two trust-region problem. Consider the following program,

$$Q^2P \qquad \min\left\{x^tQ_0x + 2b_0^tx - a_0 \mid x^tQ_kx + 2b_k^tx - a_k \le 0, 1 \le k \le m\right\}.$$

Since this program is clearly general enough to include *2-TRS* as a special case, it can exhibit duality gaps unless the objective function and the constraints are convex. We can nevertheless state standard sufficient conditions for optimality, characterizing cases where no duality gap occurs.

Lemma 3.4.1 *Assume that a pair of vectors* $x \in \mathbb{R}^n$*, feasible, and* $\lambda \in \mathbb{R}^m$*, nonnegative, satisfy the following conditions.*

$$(Q_0 + \textstyle\sum_{i=1}^m \lambda_i Q_i)x = -b_0 - \lambda^t b \qquad \textit{(stationarity)},$$

$$\lambda(x^tQ_ix + 2b_i^tx - a_i) = 0, \qquad 1 \le i \le m \qquad \textit{(complementarity)},$$

$$(Q_0 + \textstyle\sum_{i=1}^m \lambda_i Q_i) \succeq 0 \qquad \textit{(strengthened second-order)}.$$

Then x *solves* Q^2P*. Moreover, if* $(Q_0 + \sum_{i=1}^m \lambda_i Q_i) \succ 0$*, then* x *is the unique minimizer.*

Proof: The proof is similar to Lemma 2.1.7. □

From the Lagrangean dual, as in the Poljak, Rendl and Wolkowicz [52] recipe, or by introducing the now familiar vector $y = (1\ x)^t$, we can homogenize to get Q^2P in pure quadratic form,

$$Q^2P \qquad \min\left\{y^tP_0y \mid y^tE_{00}y = 1, y^tP_ky \le a_k, 1 \le k \le m, y \in \Re^{n+1}\right\},$$

where

$$P_k = \begin{bmatrix} 0 & b_k^t \\ b_k & Q_k \end{bmatrix}, \qquad 0 \leq k \leq m.$$

From this primal we derive the Lagrangean dual,

$$\textit{Dual-}Q^2P \qquad \max\left\{-\mu - \lambda^t a \mid P_0 + \mu E_{00} + \sum_{i=1}^{m} \lambda_i P_i \succeq 0, \lambda \geq 0\right\}.$$

We can also rewrite the primal in matrix form using $Y = yy^t$, as we did in the previous chapters, and drop the rank one condition to obtain the semidefinite relaxation,

$$\textit{PSDP} \qquad \min\left\{\langle P_0, Y\rangle \mid \langle E_{00}, Y\rangle = 1, \langle P_k, Y\rangle \leq a_k, 1 \leq k \leq m, Y \succeq 0\right\}.$$

We can now generalize some of the results obtained for *2-TRS*, using the same projector map and the same definitions of feasible sets. We recall the necessary definitions. The feasible set of Q^2P,

$$\hat{F} := \{x | x^t Q_k x + 2b_k^t x \leq a_k, 1 \leq k \leq m\};$$

the feasible set of *PSDP*,

$$\tilde{F} := \{Y | \langle P_k, Y\rangle \leq a_k, 1 \leq k \leq m\};$$

and the projector map,

$$P_R(\begin{bmatrix} a & x^t \\ x & X \end{bmatrix}) = x.$$

Lemma 3.4.2 *Suppose that Y is a feasible solution of PSDP. The projected vector, $x = P_R(Y)$, is then feasible for all convex constraints of Q^2P. Moreover, if the feasible set of Q^2P is convex, then $\hat{F} = P_R(\tilde{F})$.*

Proof: The proof follows from Lemma 1.4.2 applied to each convex constraint in turn. □

From the feasibility of the projected vector, we can establish some relations between optimality of Q^2P and of the semidefinite relaxation *PSDP*.

Lemma 3.4.3 *Suppose that Q^2P has a convex feasible set and that Y, an optimal solution of PSDP, is rank one, then $x = P_R(Y)$ solves Q^2P.*

Alternatively, suppose that Q^2P is a convex program and that Y is an optimal solution of PSDP, then the projection $x = P_R(Y)$ is an optimal solution for Q^2P. Moreover, if the objective function is strictly convex, Y is rank one.

Proof: The first result follows from Lemma 1.4.4. The second one, from Lemma 1.4.3. □

This takes care of the convex case and of some non-convex cases (those without any duality gap). When the constraints are convex but the objective is arbitrary, we can move along the nullspace of the Lagrangean as we did in the *2-TRS* case until we hit one of the constraints. This is possible since the the first column of the semidefinite relaxation is feasible for Q^2P. This nullspace-restricted step improves the objective value even if it does not lead to an optimal solution.

Lemma 3.4.4 *If the semidefinite primal optimal solution Y is not rank one, let $\tilde{x} = P_R(Y)$, the first column of Y. Then $x = \tilde{x} + \bar{x}$, where $\bar{x}$ is chosen in $\mathcal{N}(Q_0 + \sum \lambda_i Q_i + \mu E_{00})$, the nullspace of the Lagrangean, and so that x is feasible, will improve the primal objective value of Q^2P.*

Proof: Since $(Q_0 + \sum \lambda_i Q_i + \mu E_{00})\bar{x} = 0$, we can pre-multiply by $\bar{x}$ and obtain

$$\begin{aligned} 0 &= \bar{x}^t(Q_0 + \sum \lambda_i Q_i + \mu E_{00})\bar{x} \\ &= \bar{x}^t Q_0 \bar{x} + \bar{x}^t(\sum \lambda_i Q_i + \mu E_{00})\bar{x}, \end{aligned}$$

where the second term is nonnegative since the constraints are convex. Therefore $\bar{x}Q_0\bar{x} \leq 0$ and

$$x^t Q_0 x = (\tilde{x} + \bar{x})^t Q_0 (\tilde{x} + \bar{x}) \leq \tilde{x}^t Q_0 \tilde{x},$$

so that moving in the nullspace improves the objective value. □

We have improved the objective function but cannot guarantee optimality in the non-convex case. There is an interesting avenue to explore here, the addition of another constraint, a trust-region, around our best current solution, excluding the previous stationary point to which the algorithm had converged. This, on a few examples, seems promising.

Recall Example 2.5.5 where the semidefinite relaxation produced a strictly interior point $(5.48\ .3)^t$, which failed to satisfy complementarity. We moved along the nullspace of the Lagrangean to the boundary $(5.9925\ .3)^t$, as illustrated in Figure 3.4, a blow-up of Figure 2.5, with the first additional trust-region.

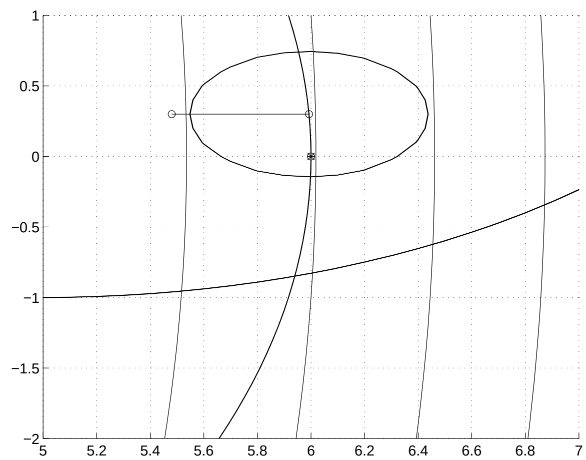


Figure 3.3: Nullspace-restricted step of Example 2.5.5.

The constructed trust-region excludes the unwanted stationary point and another iteration improves the solution. This can then be repeated. The following table illustrates these iterations, converging, albeit slowly, to the optimal $(6\ 0)^t$, from the best point produced by the relaxation, after the nullspace-restricted step.

Iter	$(x_1, x_2)^t$	$(\lambda_1, \lambda_2)^t$	$(d_1, d_2)^t$
1	(+5.9925 +0.3000)	(+1.2000 +8.8000)	(-0.5125 +0.0000)
2	(+5.9947 +0.0376)	(+0.0000 +8.9987)	(+0.0022 -0.2624)
3	(+5.9891 +0.0047)	(+0.0000 +8.9991)	(-0.0055 -0.0329)
4	(+5.9890 +0.0006)	(+0.0000 +8.9982)	(-0.0001 -0.0041)
5	(+5.9890 +0.0001)	(+0.0000 +8.9982)	(-0.0000 -0.0005)
6	(+5.9890 +0.0000)	(+0.0000 +8.9982)	(-0.0000 -0.0001)
7	(+6.0000 +0.0000)	(+0.0000 +8.9982)	(-0.0110 +0.0000)
8	(+6.0000 +0.0000)	(+0.0000 +9.0000)	(-0.0000 +0.0000)

This is barely scratching the surface of what can be done with this approach and is not meant as a proof that the additional constraints guarantees convergence, especially since the right choice of radius for the additional trust-region has not been found. But the example suggests that it might be possible to close the gap between the relaxation and the original problem, at least in some cases.

The reasons for the success of this approach are certainly related to Lemma 2.2.1. The additional constraint, a trust-region constructed to exclude the spurious stationary point, convexifies the Lagrangean and reduces the gap between our convex primal-dual approach and the original non-convex problem. We will not pursue this any further as we are restricting most of our results to the convex case.

We now have all the tools required to solve quadratically constrained convex quadratic programs. We now describe how to recursively attack more general programs.

3.5 Quadratically constrained quadratic programming

Now that a reasonable subproblem is defined and its solution is known to be useful, we combine it to our previous work on semidefinite relaxations to fully describe the SQ^2P approach. For most of the details, we will again restrict ourselves to convex programs but will indicate some extensions we have been able to make for the non-convex case.

The original problem under study is

$$NLP \qquad \min\left\{f(x) \mid g(x) \leq 0, x \in \mathbb{R}^n\right\}.$$

At some point $x^{(k)}$, possibly infeasible, we expand every function by second-order Taylor polynomials and construct the subproblem

$$\begin{array}{llcll} NLP\text{-}Q^2P & \min & q_0(d) & = & \nabla f(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 f(x^{(k)}) d \\ & \text{s.t.} & q_i(d) & = & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^t d + \frac{1}{2} d^t \nabla^2 g_i(x^{(k)}) d \leq 0, \qquad 1 \leq i \leq m \\ & & & & d^t d \leq \delta^2. \end{array}$$

We added a trust-region to guarantee a bounded subproblem, in cases of non-convex objective functions.

This problem is in the same form as every other quadratic problem we considered. Homogenization, obtained by adding a component to the vector d, together with the constraint $d_0^2 = 1$, allow the semidefinite relaxation,

$$PSDP \qquad \min\left\{\langle P_0, Y\rangle \mid \langle E_{00}, Y\rangle = 1, \langle P_i, Y\rangle \leq a_i, 1 \leq i \leq m, \langle P_I, Y\rangle \leq \delta^2, Y \succeq 0\right\},$$

where

$$P_0 = \begin{bmatrix} 0 & \nabla f(x^{(k)})^t & 0 \\ \nabla f(x^{(k)}) & \nabla^2 f(x^{(k)}) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; P_i = \begin{bmatrix} 0 & \nabla g_i(x^{(k)})^t & 0 \\ \nabla g_i(x^{(k)}) & \nabla^2 g_i(x^{(k)}) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; a_i = -2h_i(x^{(k)}),$$

and where E_{00} and P_I have their usual definitions,

$$E_{00} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, P_I = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}.$$

The dual program is

$$\text{DSDP} \qquad \max\Big\{-\mu - \lambda^t a \mid P_0 + E_{00} + \sum_{i=1}^m P_i + P_I \succeq 0, \lambda \geq 0\Big\}.$$

Solving the above pair, in the case of convex *NLP*, is enough since, as we have seen, the first column is optimal for the quadratic approximation. But, in general, we need an appropriate merit function to ensure sufficient decrease at each step and guarantee global convergence of the algorithm, whether we use a line search or a trust-region strategy.

The choice of merit function for *SQP* algorithms varies considerably. For infeasible iterates there is a need to balance improvement in the objective function and movement towards feasibility. The description of the following merit functions owes much to the survey papers of Boggs and Tolle [8] and of Stoer [66]. Where it appears, the constant η is positive and increasing at each iteration of the algorithm. The rules for increase vary from one implementation to the next and are described in the referenced papers.

The classical choice is the quadratic penalty function

$$\varphi(x) = f(x) + \sum_{i \in I} g_i(x)^2,$$

where the index set includes only violated constraints, i.e., $I := \{i | g_i(x) > 0\}$. A better choice, for equality constrained, is the augmented Lagrangean,

$$\varphi(x) = f(x) + h(x)^t \lambda + \eta ||h(x)||^2.$$

It was first proposed by Fletcher [19] and then used as a merit function by Powell and Yuan [54]. An analogue for inequality constrained optimization was introduced by Rockafellar [58],

$$\varphi(x) = f(x) + 1/(4\eta) \sum_{i=1}^m [\theta(\lambda_i + 2\eta g_i(x))^2 - \lambda_i^2],$$

where $\theta(t) = \max\{0, t\}$ extracts only the violated constraints. Shittkowski [60],[61] developed a related form,

$$\varphi(x) = f(x) + \sum_{i \in I}[\lambda_i g_i(x) + \frac{1}{2}\eta g_i(x)^2] + 1/(2\eta)\sum_{i \notin I}\lambda_i^2,$$

where the multipliers, interestingly, are those of *QP*, not those obtained from the least-square problem. The index set is defined as

$$I := \{i | g_i(x) \geq -\frac{\lambda_i}{\eta}\},$$

and includes all violated constraints and no "safely" satisfied constraints.

Another avenue uses the l_1 exact penalty function, which, for equality constrained problems yields

$$\varphi(x) = f(x) + \eta ||h(x)||_1,$$

and was first suggested by Han [27] as a merit function.

Finally, because the solutions of *NLP*-Q^2P tend to points satisfying at least the first-order conditions of *NLP*, the merit function can be

$$\varphi(x) = ||\nabla \mathcal{L}(x, \lambda)||^2 + ||h(x)||^2,$$

a particularly appropriate choice for convex programs as described in [37].

We will come back, briefly, to the merit function when we investigate convergence of the algorithm but we first complete its description. After solving the Q^2P subproblem for a direction $d \neq 0$, the next iterate is obtained by $x^{(k+1)} = x^{(k)} + d$. This new point serves for the expansion of a new problem by second-order polynomials and we iterate until the subproblem yields $d = 0$. As with any trust-region based algorithm, we adjust the trust-region radius according to the ratio of predicted improvement to actual improvement. At the end, we have a solution satisfying both first and second-order conditions of *NLP*. Somewhat more formally, here is the SQ^2P algorithm.

SEQUENTIAL QUADRATICALLY CONSTRAINED PROGRAMMING ALGORITHM

$\boldsymbol{SQ^2P}(f, \nabla f, \nabla^2 f, g_i, \nabla g_i, \nabla^2 g_i, x^{(0)})$

do

$Y \in \text{argmin}\{\langle P_0, Y\rangle : \langle P_i, Y\rangle \leq a_i, \langle E_{00}, Y\rangle = 1, Y \succeq 0\}$

$(\mu, \lambda^{(k+1)}) \in \text{argmax}\{-\mu - \sum \lambda_i a_i : P + \sum \lambda P_i + \mu E_{00} \succeq 0, \lambda \geq 0\}$

$d = P_R(Y)$

$x^{(k+1)} = x^{(k)} + d$

$r^k = \frac{\varphi(x^{(k)}) - \varphi(x^{(k+1)})}{q_0(x^{(k)}) - q_0(x^{(k+1)})}$

if $(r^k < \frac{1}{4})$

$\delta = \delta/4$

elseif $(r^k > \frac{3}{4})$ and $||x^{(k+1)} - x^{(k)}|| = \delta$

$\delta = 2\delta$

fi

$k = k + 1$

while $(||d|| > \varepsilon)$

Find maximal $d \in \mathcal{N}(\nabla\mathcal{L})$ such that $g(x^{(k)} + d) \leq 0$

$x^{(k)} = x^{(k)} + d$

return$(x^{(k)}, \lambda^{(k)})$

3.6 Convergence of SQ^2P

We now investigate the convergence of an iterative algorithm developped within the SQ^2P framework.

For the asymptotic convergence rate, we will make a simplifying assumption: When $x^{(k)}$ is close to x^*, the active constraints of the Q^2P subproblem are the same as the active constraints of *NLP*. (For more detail on this assumption, see Boggs and Tolle [8].) That we have identified the active, and therefore the inactive constraints for *NLP* at x^*, allows us to ignore inactive constraints and change the active constraints to equalities. Therefore, under this assumption, we need consider only the equality-constrained program

$$NEP \qquad \min\Big\{f(x)\ \Big|\ h(x)=0, x\in\mathbb{R}^n\Big\}.$$

We rewrite the stationarity condition of the Lagrangean as

$$\nabla\mathcal{L}(x^{(k)},\lambda^{(k)}) = \begin{bmatrix} \nabla_x\mathcal{L}(x^{(k)},\lambda^{(k)}) \\ \nabla_\lambda\mathcal{L}(x^{(k)},\lambda^{(k)}) \end{bmatrix} = \begin{bmatrix} \nabla f(x^{(k)}) + h'(x^{(k)})^t\lambda \\ h(x^{(k)}) \end{bmatrix} = 0.$$

We have described in detail, in a previous section, how a higher-order Newton method can be used to solve a system of nonlinear equations as the one above. From the definition of $\mathcal{L}$, a second-order Newton step can be written as a nonlinear system of equation in δ_x and δ_λ. The manner in which we write it has little to do with a method of solution. It has everything to do with the comparison we wish to make between three systems of equations: From a second-order Newton method, from the SQ^2P step and from the standard SQP step.

First, here is the second-order Newton step,

$$\begin{bmatrix} \sum\lambda_i\nabla h_i(x^{(k)}) + \big(\nabla^2 f(x^{(k)}) + \sum\lambda_i\nabla^2 h_i(x^{(k)})\big)\,\delta_x + H_3(\delta_x,\delta_\lambda) \\ h'(x^{(k)})\delta_x + \frac{1}{2}\delta_x^t h''(x^{(k)})\delta_x \end{bmatrix} = \begin{bmatrix} -\nabla f(x^{(k)}) \\ -h(x^{(k)}) \end{bmatrix},$$

where we have grouped the third-order derivatives of f and h under the name H_3. We can contrast this step to stationarity of the Lagrangean of *NLP*-Q^2P,

$$\begin{bmatrix} \sum\lambda_i\nabla h_i(x^{(k)}) + \big(\nabla^2 f(x^{(k)}) + \sum\lambda_i\nabla^2 h_i(x^{(k)})\big)\,\delta_x \\ h'(x^{(k)})\delta_x + \frac{1}{2}\delta_x^t h''(x^{(k)})\delta_x \end{bmatrix} = \begin{bmatrix} -\nabla f(x^{(k)}) \\ -h(x^{(k)}) \end{bmatrix},$$

and to stationarity of the Lagrangean of the QP subproblem or, equivalently, of a first-order

Newton step,

$$\begin{bmatrix} \sum \lambda_i \nabla h_i(x^{(k)}) + \left(\nabla^2 f(x^{(k)}) + \sum \lambda_i \nabla^2 h_i(x^{(k)})\right) \delta_x \\ h'(x^{(k)})\delta_x \end{bmatrix} = \begin{bmatrix} -\nabla f(x^{(k)}) \\ -h(x^{(k)}) \end{bmatrix}.$$

The main difference between the stationarity of *NLP-Q^2P* and a second-order Newton's method lies in the third derivative terms missing in the former. But the second-order terms related to the curvature of the constraints are present and this is where we expect *SQ^2P* to overtake *SQP*, namely when the original problem has highly curved constraints. Viewed differently, the *NLP-Q^2P* subproblem produces a first-order step towards stationarity and a second-order step towards feasibility.

The asymptotic q-quadratic convergence rate follows from stationarity of *NLP-Q^2P*, as expressed above, and from the convergence of a Newton's method of order t, as given in Proposition 3.2.1.

For global convergence of the algorithm from an arbitrary starting point we use the following merit function

$$\varphi(x^{(k)}, \lambda^{(k)}) = \frac{1}{2} \left\| \begin{matrix} \nabla \mathcal{L}(x^{(k)}, \lambda^{(k)}) \\ g^+(x^{(k)}) \end{matrix} \right\|_2^2,$$

where

$$g_i^+(x^{(k)}) = \begin{cases} g_i(x^{(k)}), & \text{if } g_i(x^{(k)}) > 0; \\ 0, & \text{if } g_i(x^{(k)}) \leq 0. \end{cases}$$

The derivative, in the direction d, satisfies

$$d^t \nabla \varphi(x^{(k)}, \lambda^{(k)}) = d^t \nabla^2 \mathcal{L}(x^{(k)}, \lambda^{(k)}) + \sum d^t g_i^+(x^{(k)}) \nabla g_i(x^{(k)}),$$

which, since the solution of *NLP-Q^2P*, d satisfies the system

$$\begin{aligned} \nabla^2 \mathcal{L}(x^{(k)}, \lambda^{(k)})d + \nabla \mathcal{L}(x^{(k)}, \lambda^{(k)}) &= 0 \\ \frac{1}{2} d^t \nabla^2 g_i(x^{(k)})d + \nabla g_i(x^{(k)})d + g_i(x^{(k)}) &\leq 0, \end{aligned}$$

implies $d^t \nabla \varphi(x^{(k)}, \lambda^{(k)}) \leq 0$, i.e., d is a descent direction for the merit function. Notice that this descent property does not rely on convexity. It therefore applies to general nonlinear programs.

3.7 Conclusion

The SQ^2P approach solves convex programs to a vector satisfying first and second-order conditions with a convergence rate at least as high as standard SQP; higher if the constraints are highly curved. The following example illustrates how the two methods compare. To simplify the presentation and highlight the main difference between the algorithms, a full step was taken at each iteration of SQP and, in the case of SQ^2P, the trust-region was chosen large enough never to be binding.

Example 3.7.1 *Illustrative comparison of SQP and* SQ^2P.

$$\min\left\{-x_1 - x_2 \mid x_1^3 - x_2 \leq 0,\ x_1^3 + x_2^2 - 1 \leq 0\right\}$$

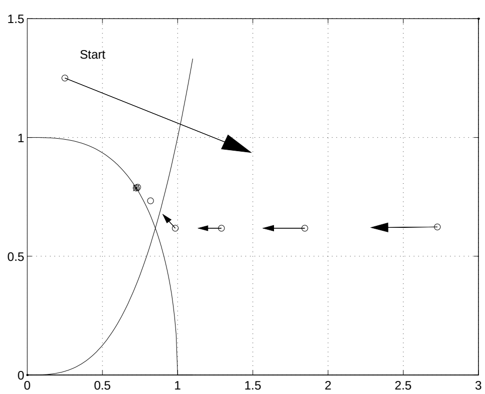


Figure 3.4: Iterations of SQP on Example 3.7.1, from initial point $(\frac{1}{4}\ \frac{5}{4})^t$. *As the first iteration demonstrates, the direction given by the QP subproblem can be poor.*

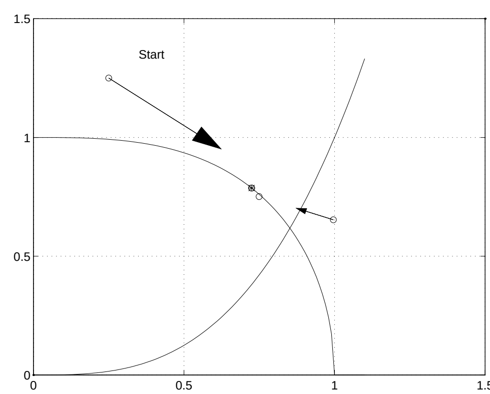


Figure 3.5: Iterations of SQ^2P *on the same example. The horizontal scale is changed to highlight the value of the direction provided by the* Q^2P *subproblem.*

The strength of the algorithm lies in the uses it makes of the curvature of the constraints while remaining very simple to implement. This is in striking contrast to most algorithms that take advantage of second-order information. The subproblem provides a better balance between primal and dual iterates. This balance, in turn, allows a faster convergence rate.

The next step in this direction is obviously to extend the algorithm to non-convex problems. As it is, we only guarantee an approximation to the optimal solution in the non-convex case since the results concerning descent direction and convergence to points satisfying first and second-order conditions, even if they apply to arbitrary objective functions, only concern *NLP*-Q^2P, not its semidefinite relaxation. And since these two are different unless *NLP* is convex, it does not immediately follow that the solution produced by the relaxation also is a descent direction or that, if it is, the limit points of the algorithm satisfy the optimality conditions. But a convex set with an arbitrary objective, as in the *2-TRS* case should be tractable by this approach. Some preliminary results seem to indicate that this is indeed the case, but much more work remains to be done.

We have completely neglected the possibility of an inconsistent subproblem. This has been the cause of much concern in the standard *SQP* case but has known solutions, the simplest of which is to take a steepest-descent step on the merit function whenever the subproblem is inconsistent. This is justified by the expectation that the subproblem can only be inconsistent when the iterates are unreasonably far from a solution and a few steepest-descent steps will forever remedy the problem.

The line searches also need to be considered for future research. There are two of them. There would be three if we had not used a trust-region approach. The inner line search is done within the semidefinite solver to remain in the cone of positive definite matrices. Then there is the nullspace-restricted step to improve the solution in the case of non-convex objective. And finally, as an alternative to a trust-region strategy, there is the traditional search along the direction provided by the subproblem. There may be relations between these searches to be analyzed and exploited. And the alternative "trust-region or line search" is worthy of further research. Also, the last line search can be done either with the original constraints or with the quadratic approximations. In the former case, we may guarantee feasible estimates, in the latter, we may improve the convergence rate, at least in some cases.

Chapter 4

Semidefinite solver

The work we have done to solve nonlinear programs assumed a solver for semidefinite linear programs, much in the same way that SQP requires a QP solver. We have, until now, considered this solver as a black box into which we stored the program and from which we retrieved an optimal solution. This was appropriate for the exposition of the work and was, in fact, how the implementation of SQ^2P was initially carried out. We relied on the code of Rendl [56].

There is little to be gained by a special-purpose code since, for our subproblem, there is no structure that can simplify or reduce the size of the problem, as there is, for example, when semidefinite programming is used to solve combinatorial problems like quadratic assignment or graph partitioning. But, for the sake of completeness, we sketch here a primal-dual approach that can be used to solve a pair of semidefinite linear programs. We closely follow [30] to describe this implementation of an interior point algorithm.

The original primals we considered throughout in $\mathbb{R}^n$ can be stated as

$$\mathrm{P}_E \qquad \min\left\{x^tQ_0x + 2b_0^tx + a_0 \;\middle|\; x^tQ_ix + 2b_i^tx + a_i, 1 \le i \le m, x^tx \le \delta^2\right\}.$$

After homogenization we move into the semidefinite cone and get the following relaxation

$$\mathrm{P}_K \qquad \min\left\{\langle P_0, Y\rangle \;\middle|\; \langle E_{00}, Y\rangle = 1, \langle P_i, Y\rangle \le -a_i, 1 \le i \le m+1, Y \succeq 0\right\}.$$

Where we made the simplifying notational changes,

$$a_{m+1} = -\delta^2,\ Q_{m+1} = I,\ P_i = \begin{bmatrix} 0 & b_i^t \\ b_i & Q_i \end{bmatrix}, 1 \leq i \leq m+1,\ b_{m+1} = 0,\ E_{00} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

We introduce slack variables z_i in order to get a pair of symmetric primal-dual programs; the first of which now reads

$$\mathrm{P}_K \qquad \min\Big\{\langle P_0, Y\rangle \ \big|\ \langle E_{00}, Y\rangle = 1, \langle P_i, Y\rangle + z_i = -a_i, 1 \leq i \leq m+1, z \geq 0, Y \succeq 0\Big\}.$$

It is clear that each primal semidefinite program considered in the previous chapters can be formulated in this manner. To find the correct dual, we consider the min-max program,

$$\min\Big\{\max\Big\{\langle P_0, Y\rangle + \mu(\langle E_{00}, Y\rangle - 1) + \sum_1^{m+1} \lambda_i(\langle P_i, Y\rangle + z_i + a_i) \mid \lambda \in \mathbb{R}^{m+1}\Big\} \mid z \geq 0, Y \succeq 0\Big\}.$$

This reduces to our primal since, for the maximization to be bounded, we get back primal feasibility. We get the dual by reversing the max and min.

$$\max\Big\{\min\Big\{\langle P_0 + \mu E_{00} + \sum_1^{m+1} P_i, Y\rangle - \mu + \lambda^t z + \lambda^t a \mid z \geq 0, Y \succeq 0\Big\} \mid \lambda \in \mathbb{R}^{m+1}\Big\}.$$

The boundedness of the inner minimization implies

$$\langle P_0 + \mu E_{00} + \sum_1^{m+1} P_i, Y\rangle \geq 0, \quad \text{and} \quad \lambda^t z \geq 0,$$

which, since $Y \succeq 0$ and $z \geq 0$, reduces to $P_0 + \mu E_{00} + \sum_1^{m+1} P_i \succeq 0$ and $\lambda \geq 0$. We therefore get, after adding slack variable Z, the dual program

$$\mathrm{D}_K \qquad \min\Big\{-\mu + \lambda^t a \ \big|\ P_0 + \mu E_{00} + \sum_1^{m+1} P_i = Z, Z \succeq 0, \lambda \geq 0\Big\}.$$

We now have the symmetric pair of primal and dual programs we required. For an interior-point method, we need a measure of the duality gap which we get by computing

$$\begin{aligned}\langle P_0, Y\rangle - (-\mu + \lambda^t a) &= \langle Z - \mu E_{00} - \sum_1^{m+1} P_i, Y\rangle + \mu + \sum_1^{m+1} \lambda_i(\langle P_i, Y\rangle + z_i) \\ &= \langle Z, Y\rangle - \langle \mu E_{00} + \sum_1^{m+1} P_i, Y\rangle + \mu + \langle \sum_1^{m+1} P_i, Y\rangle + \lambda^t z \\ &= \langle Z, Y\rangle + \lambda^t z.\end{aligned}$$

Since we assumed throughout that a strictly interior point was feasible, there is no duality gap. Therefore $\langle Z, Y\rangle + \lambda^t z = 0$ and, since $Z, Y \succeq 0$, and $z, \lambda \geq 0$, we also have that both products $\langle Z, Y\rangle$, and $\lambda^t z$ are nonnegative. This leads to the complementarity equations we required, namely

$$ZY = 0 \quad \text{and} \quad \lambda \circ z = 0,$$

where $(\circ)$ is denotes the Hadamard product.

To derive an interior-point method from these equations, we now introduce a barrier parameter $\gamma \to 0$ and rewrite the complementarity equation as $\lambda^t z = \gamma$ and $\langle Z, Y\rangle = \gamma$. From which we conclude that, for a given solution Z, Y, z, λ, we can compute

$$\gamma = \frac{1}{n+m+1}\{\langle Z, Y\rangle + z^t \lambda\}.$$

We call this a barrier parameter since the derivation of the above equations for primal and dual feasibility as well as complementarity can be done using a so-called dual-barrier program.

$$\min\Big\{-\mu + \lambda^t a + \gamma(\log\det Z + \log\lambda) \;\Big|\; \mu E_{00} + \sum_1^{m+1} P_i + P_0 = Z, Z \succeq 0, \lambda \geq 0\Big\},$$

where $\gamma \geq 0$ is the barrier parameter. A sequence of such parameters converging to zero generates a sequence of $\mu_\gamma, \lambda_\gamma$ converging to a fixed point, solving the dual semidefinite program. The

Lagrangean associated to a specific value of γ is

$$\mathcal{L}(\mu, \lambda, Y, Z, z) = -\mu + \lambda^t a + \gamma(\log\det Z + \log\lambda) + \langle -Z + \mu E_{00} + \sum_1^{m+1} P_i + P_0, Y\rangle.$$

The stationarity condition for this Lagrangean, namely $\nabla\mathcal{L} = 0$, reduces to primal feasibility, dual feasibility and complementary-slackness, but depends on the barrier parameter γ.

We expand the stationarity condition as

$$F_\gamma := \begin{cases} F_d & := & Z - P_0 - \mu E_{00} + \sum_1^{m+1} P_i & = & 0 \\ F_e & := & 1 - \langle E_{00}, Y\rangle & = & 0 \\ (F_n)_i & := & z_i + \langle P_i, Y\rangle + a_i & = & 0 \\ F_{cs1} & := & ZY - \gamma I & = & 0 \\ F_{cs2} & := & z \circ \lambda - \gamma e & = & 0 \end{cases}$$

The interior-point approach computes, for a given value of γ, the damped Newton step $\Delta s = (\Delta\mu, \Delta\lambda, \Delta Y, \Delta Z, \Delta z)$ satisfying

$$F_\gamma(\mu, \lambda, Y, Z, z) + \nabla F_\gamma(\mu, \lambda, Y, Z, z)\Delta s = 0.$$

Differentiation yields

$$\begin{array}{lcl} \Delta Z - \Delta\mu E_{00} - \sum_1^{m+1} \Delta\lambda_i P_i & = & -F_d \\ -\langle E_{00}, \Delta Y\rangle & = & -F_e \\ \Delta z_i + \langle P_i, \Delta Y\rangle & = & (-F_n)_i \\ \Delta Z Y + Z \Delta Y & = & -F_{cs1} \\ \Delta z \circ \lambda + Z \circ \Delta\lambda & = & -F_{cs2} \end{array}$$

This system can be solved for $\Delta\mu, \Delta\lambda$, which are then substituted back to obtain $\Delta Y, \Delta Z$. The new value for Y may not be symmetric, in which case symmetry-inducing transformation such as $Y = (Y + Y^t)/2$ is used. Finally a step is taken in that direction provided it maintains strict feasibility,

$$Y = Y + \Delta Y,\ Z = Z + \Delta Z,\ \lambda = \lambda + \Delta\lambda,\ \mu = \mu + \Delta\mu.$$

Chapter 5

Semidefinite programming for continuous optimization

Semidefinite programming has only recently enjoyed the attention of the mathematical programming community. Even if some of its roots date as far back as the discipline as a whole – A monograph of Berman [5] refers to papers written in the sixties, barely ten years after the birth of the simplex method for linear programming (Dantzig [13]) and of the optimality conditions for nonlinear programming (Kuhn and Tucker [33]). – its name and most of its successes are fairly recent. The name itself seems to have appeared in the nineties (Alizadeh, Haeberly and Overton [1], Helmberg, Poljak, Rendl and Wolkowicz [29], Nesterov and Nemirovsky [47]), even if some researchers in the field, until recently, preferred a different name. A paper that has circulated for the past two years under the name *Positive Definite Programming* (Vandenberghe and Boyd [68]),a name justified by the algorithmic approach used to solve such problems has just been published under the title *Semidefinite Programming* [69]. The name seems now firmly entrenched.

The successes are due in no small part to Nesterov and Nemirovsky [47] who provided a unifying framework for polynomial-time algorithms and to Goemans and Williamson [26] who radically improved, by means of a semidefinite relaxation, the guaranteed bounds for max-cut and max-2sat, both classical problems of combinatorial optimization. The use of this kind of relaxation has multiplied thereafter in discrete optimization (Lovász and Schriver [35], Poljak and Wolkowicz [53]). Work in continuous optimization (Stern and Wolkowicz [63]) has also led to interesting results, most notably a very fast algorithm solving the trust-region subproblem, an archetypal problem of nonlinear optimization.

The work presented here aimed above all to collect disparate results concerning optimization of quadratic surfaces over ellipsoids. And then to show how semidefinite programming first provides a simple framework to describe such problems – contrast the two given proofs of the strong duality of *TRS*– and second produces very simply-stated algorithms. Especially when these algorithms are compared to classical algorithms that use curvature information. Of course the semidefinite programs would not be easy to solve if it were not for the very recent developments of interior-point methods.

Our work is, in some sense very classical since we extend old and trusted methods like *SQP* and trust-regions to look for points satisfying both Karush-Kuhn-Tucker and second-order conditions. We do this by modifying the standard *SQP* subproblem. But the tools required to solve this subproblem owe much to recent work, both for the lifting of the problem into a different space and for the interior-point methods of solution.

The approach was restricted to convex programs where the implementation was concerned and this offers a possible path for future work. The choice of subproblems for non-convex programs might be different, relying more on the Lagrangean or other convexity inducing functions. The relation between the original feasible set and the lifted set also offers avenues of research and might lead to different relaxations. The implementation itself needs to be more robust and the best choice of interior-point algorithm is unclear. A dual-only approach might be more appropriate since the dimension of the duals considered will often be much smaller than the dimension of the primals. All of these provide much food for thought and will be considered, in time.

Bibliography

[1] F. Alizadeh, J-P.A. Haeberly, and M.L. Overton. A new primal-dual interior-point method for semidefinite programming. Technical report, Courant Institute of Mathematical Sciences, 1994. Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, Snowbird, Utah, June, 1994.

[2] M. S. Bazaraa and C. M. Shetty. *Nonlinear Programming : Theory and Algorithms.* Wiley and Sons, 1979.

[3] E.M.L. Beale. Numerical methods. In J. Abadie, editor, *Nonlinear Programming.* North-Holland, Amsterdam, 1967.

[4] A. Ben-Israel and T.N.E. Greville. *Generalized Inverses: Theory and Applications.* Wiley-Interscience, 1974.

[5] A. Berman. *Cones, Matrices and Mathematical Programming.* Springer-Verlag, Berlin, New York, 1973.

[6] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Mass., 1995.

[7] M. J. Best and K. Ritter. *Quadratic Programming Active Set Analysis and Computer Programs.* University of Waterloo, 1995.

[8] P.T. Boggs and J.W. Tolle. Sequential quadratic programming. *Acta Numerica*, pages 1–51, 1995.

[9] R.H. Byrd, R.B. Schnabel, and G.A. Schultz. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM J. Numer. Anal.*, 22:47–67, 1985. Also available as Technical Report CU-CS-216-82, Department of Computer Science, University of Colorado, Boulder, CO.

[10] M.R. Celis, J.E. Dennis Jr., and R.A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In *Proceedings of the SIAM Conference on Numerical Optimization, Boulder, CO*, 1984. Also available as Technical Report TR84-1, Department of Mathematical Sciences, Rice University, Houston, TX.

[11] T. F. Coleman and A. R. Conn. Nonlinear programming via an exact penalty function: Asymptotic analysis. *Math. Prog.*, 24:123–136, 1982.

[12] A. R. Conn. An efficient second order method to solve the (constrained) minimax problem. Technical Report CORR 79-5, Dept. of Combinatorics and Optimization, University of Waterloo, Ontario, Canada, 1979.

[13] G. B. Dantzig. Programming in a linear structure. *Econometrica*, 17, 1949.

[14] J.E. Dennis Jr. and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996. Originally published: Englewood Cliffs, N.J. : Prentice-Hall.

[15] S. Di and W. Sun. A trust region method for conic model to solve unconstrained optimization. *Manuscript*, 1996.

[16] L.C.W. Dixon, S.E. Hersom, and Z.A. Maany. Initial experience obtained solving the low thrust satellite trajectory optimisation problem. Technical Report T.R. 152, The Hatfield Polytechnic Numerical Optimization Centre, 1984.

[17] N.R. Draper. Ridge analysis of surface response. *Technometrics*, 5:469–479, 1959.

[18] M. El-Alem. A global convergence theory for the cdt trust-region algorithm for constrained optimization. *SIAM Journal of Numerical Analysis*, 28:266–290, 1991.

[19] R. Fletcher. A class of methods for nonlinear programming. In F.A. Lootsma, editor, *Numerical Methods for Nonlinear Optimization*. Academic Press, New York, 1972.

[20] R. Fletcher. *Practical Methods of Optimization, vol. 1, Unconstrained Optimization*. John Wiley & Sons, New York, NY, 1980.

[21] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, NY, second edition, 1987.

[22] G.E. Forsythe and G.H. Golub. On the stationary values of a second-degree polynomial on the unit sphere. *SIAM J. Applied Mathematics*, 13:1050–1068, 1965.

[23] T. Fujie and M. Kojima. Semidefinite programming relaxation for nonconvex quadratic programs. *Technical report*, 1995.

[24] D.M. Gay. Computing optimal locally constrained steps. *SIAM J. Sci. Statist. Comput.*, 2:186–197, 1981.

[25] P.E. Gill, M.A. Saunders, and W. Murray. Snopt: An sqp algorithm for large-scale constrained optimization. Technical Report NA 96-2, University of California, San Diego., 1996.

[26] M.X. Goemans and D.P. Williamson. .878-approximation algorithms for MAX CUT and MAX 2SAT. In *ACM Symposium on Theory of Computing (STOC)*, 1994.

[27] S.P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications.*, 22:297–309, 1977.

[28] M. Heinkenschlos. On the solution of a two ball trust region subproblem. Technical Report Nr.92-16, Universitat Trier, Mathenatik/Informatik, 1992.

[29] C. Helmberg, S. Poljak, F. Rendl, and H. Wolkowicz. A semidefinite programming approach to integer programming. Technical Report CORR Report 95-13, University of Waterloo, 1995.

[30] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior point method for semidefinite programming. *SIAM Journal on Optimization*, pages 342–361, 1996. URL: ftp://orion.uwaterloo.ca/pub/henry/reports/sdp.ps.gz.

[31] A.E. Hoerl. Optimal solution of many-variable equations. *Chemical engineering progress*, 55:69–78, 1959.

[32] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.

[33] H.W. Kuhn and A.W. Tucker. Nonlinear programming. *Econometrica*, 19, 1951.

[34] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[35] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[36] D.G. Luenberger. *Optimization by Vector Space Methods.* John Wiley, 1969.

[37] D.G. Luenberger. *Introduction to Linear and Nonlinear Programming.* Reading. Addison-Wesley, Massachusetts, 1973.

[38] D.G. Luenberger. *Linear and Nonlinear Programming.* Reading. Addison-Wesley, Massachusetts, second edition, 1984.

[39] S. Lyle and M. Szularz. Local minima of the trust region problem. *Journal of Optimization Theory and Applications*, 80:117–134, 1994.

[40] Z.A. Maany. A new algorithm for highly curved constrained optimization. Technical Report T.R. 161, The Hatfield Polytechnic Numerical Optimization Centre, 1985.

[41] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[42] J.M. Martinez. Local minimizers of quadratic functions on Euclidean balls and spheres. *SIAM Journal on Optimization*, 4(1):159–176, 1994.

[43] J.J. Moré. Generalizations of the trust region problem. Technical Report MCS-P349-0193, Argonne National Labs, Argonne, IL, 1993.

[44] J.J. Moré and D.C. Sorensen. Computing a trust region step. *SIAM J. Sci. Statist. Comput.*, 4:553–572, 1983.

[45] R.H. Myers and W.H. Carter. Response surface techniques for dual response systems. *Technometrics*, 15:301–317, 1973.

[46] J. C. Nash. Optimization software. *OR/MS Today*, April 1995:71, 1995.

[47] Y. E. Nesterov and A. S. Nemirovsky. *Interior Point Polynomial Algorithms in Convex Programming : Theory and Algorithms.* SIAM Publications. SIAM, Philadelphia, USA, 1994.

[48] E. R. Panier and A. L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Mathematical Programming*, 59:261–276, 1993.

[49] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. Technical report, GSIA Carnegie Mellon University, Pittsburgh, PA, 1995.

[50] M.H. Wright P.E. Gill, W. Murray. *Practical Optimization.* Academic Press, New York, London, Toronto, Sydney and San Francisco, 1981.

[51] Ji-Ming Peng and Ya-Xiang Yuan. Generalization of the trust-region subproblem. *Manuscript*, 1995.

[52] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 7:51–73, 1995.

[53] S. Poljak and H. Wolkowicz. Convex relaxations of 0-1 quadratic programming. *Mathematics of Operations Research*, 20(3):550–561, 1995.

[54] M.J.D. Powell and Y. Yuan. A recursive quadratic programming algorithm that uses differentiable penalty functions. *Math. Prog.*, 7:265–278, 1986.

[55] M.J.D. Powell and Y. Yuan. A trust region algorithm for equality constrained optimization. Technical Report DAMTP 1986/NA2, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England, 1986.

[56] F. Rendl. A matlab toolbox for semidefinite programming. Technical report, Technische Universitat Graz, Graz, Austria, 1994.

[57] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. Technical Report CORR Report 94-32, University of Waterloo, Waterloo, Canada, 1995. Accepted in Math. Progr. subject to revision.

[58] R.T. Rockafellar. Augmented lagrange multipliers functions and duality in nonconvex programming. *SIAM Journal on Control*, 12:268–285, 1973.

[59] S.A. Santos and D.C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. Technical Report TR95-20, Rice University, Houston, TX, 1995.

[60] K. Schittkowskik. The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian type line search function, Part 2: An efficient implementation with linear least squares subproblems. *Numer. Math.*, 38:115–127, 1981.

[61] K. Schittkowskik. On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function. Technical Report SOL 82-4, Department of Operations Research, Stanford University, Stanford, CA, 1982.

[62] D.C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19:409–426, 1982.

[63] R. Stern and H. Wolkowicz. Trust regions and nonsymmetric eigenvalue perturbations. *SIAM J. Matrix Analysis and Applications*, 15(3):755–778, 1994.

[64] R. Stern and H. Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optimization*, 5(2):286–313, 1995.

[65] G.W. Stewart. On infinitely many algorithms for solving equations. Technical Report TR-92-121, University of Maryland, 1993. Translation of the 1870 Schröder paper.

[66] J. Stoer. Principles of sequential quadratic programming methods for solving nonlinear programs. In K. Schittkowski, editor, *Computational Mathematical Programming*, pages 165–207. Springer-Verlag, Berlin, 1985.

[67] Pham Dinh Tao and Le Thi Hoai An. Lagrangian stability and global optimality in nonconvex quadratic minimization over euclidean balls and spheres. *Chemical engineering progress*, 55:69–78, 1959.

[68] L. Vandenberghe and S. Boyd. Positive definite programming. In *Mathematical Programming: State of the Art, 1994*, pages 276–308. The University of Michigan, 1994.

[69] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[70] R. J. Vanderbei. Loqo: An interior point code for quadratic programming. Technical report, Program in Statistics & Operations Research, Princeton University, Princeton,NJ, 1994.

[71] G.G. Vining and R.H. Myers. Combining taguchi and response surface philosophies: A dual response approach. *Journal of Quality Technology*, 22:38–45, 1990.

[72] U. Von Matt W. Gander, G. Golub. A constrained eigenvalue problem. Technical report, Institut fur Informatik ETH-Zentrum, CH-8092 Zurich Switzerland, 1988.

[73] K.A. Williamson. *A robust trust region algorithm for nonlinear programming.* PhD thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1990.

[74] R.B. Wilson. *A simplicial algorithm for concave programming.* PhD thesis, Harvard University, 1963.

[75] Y. Yuan. On a subproblem of trust region algorithms for constrained optimization. *Mathematical Programming*, 47:53–63, 1990.

[76] Y. Yuan. A dual algorithm for minimizing a quadratic function with two quadratic constraints. *Journal of Computational Mathematics*, 9:348–359, 1991.

[77] Y. Zhang. Computing a Celis-Dennis-Tapia trust-region step for equality constrained optimization. *Mathematical Programming*, 55:109–124, 1992.