SEQUENTIAL QUADRATIC PROGRAMMING METHODS[*]

Philip E. GILL[†] Elizabeth WONG[†]

UCSD Department of Mathematics
Technical Report NA-10-03
August 2010

Abstract

In his 1963 PhD thesis, Wilson proposed the first sequential quadratic programming (SQP) method for the solution of constrained nonlinear optimization problems. In the intervening 48 years, SQP methods have evolved into a powerful and effective class of methods for a wide range of optimization problems. We review some of the most prominent developments in SQP methods since 1963 and discuss the relationship of SQP methods to other popular methods, including augmented Lagrangian methods and interior methods.

Given the scope and utility of nonlinear optimization, it is not surprising that SQP methods are still a subject of active research. Recent developments in methods for mixed-integer nonlinear programming (MINLP) and the minimization of functions subject to differential equation constraints has led to a heightened interest in methods that may be "warm started" from a good approximate solution. We discuss the role of SQP methods in these contexts.

Key words. Large-scale nonlinear programming, SQP methods, nonconvex programming, quadratic programming, KKT systems.

AMS subject classifications. 49J20, 49J15, 49M37, 49D37, 65F05, 65K05, 90C30

1. Introduction

This paper concerns the formulation of methods for solving the smooth nonlinear programs that arise as subproblems within a method for mixed-integer nonlinear programming (MINLP). In general, this subproblem has both linear and nonlinear constraints, and may be written in the form

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f(x) \\ \text{subject to} & \ell \le \left\{ \begin{array}{c} x \\ Ax \\ c(x) \end{array} \right\} \le u, \end{array} \tag{1.1}$$

where $f(x)$ is a linear or nonlinear objective function, $c(x)$ is a vector of m nonlinear constraint functions $c_i(x)$, A is a matrix, and l and u are vectors of lower and upper bounds. Throughout, we assume that the number of variables is large, and that A and the derivatives of f and c are sparse. The constraints involving the matrix A and functions $c_i(x)$ will be called the *general* constraints; the remaining constraints will be called *bounds.* We assume that the nonlinear functions are smooth and that their first and second derivatives

[*] Research supported by the National Science Foundation under Grant No. DMS-0915220, and the Department of Energy under grant DE-SC0002349

[†] Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 (pgill@ucsd.edu, elwong@ucsd.edu).

are available. An *equality* constraint corresponds to setting $\ell_i = u_i$. Similarly, a special "infinite" value for ℓ_i or u_i is used to indicate the absence of one of the bounds.

The nonlinear programs that arise in the context of MINLP have several important features. First, the problem is one of a sequence of many related NLP problems with the same objective but with additional linear or nonlinear constraints. For efficiency, it is important that information from the solution of one problem be used to "warm start" the solution of the next. Second, many MINLP methods generate infeasible subproblems as part of the solution process (e.g., in branch and bound methods). This implies that infeasible constraints will occur for a significant subset of problems to be solved, which is in contrast to the situation in conventional optimization, where constraint infeasibility is considered to be a relatively unusual event, caused by an error in the constraint formulation. In mixed-integer linear programming, the phase 1 simplex method provides a reliable "certificate of infeasibility", i.e., a definitive statement on whether or not a feasible point exists. However, the question of the solvability of a set of nonconvex inequalities is NP-hard. Many optimization methods replace the solvability problem by a minimization problem in which the norm of the constraint residual is minimized. If the constraints are not convex, then this minimization problem is likely to have infeasible local minimizers regardless of the existence of a feasible point. It follows that the minimization method may terminate at a "phantom" infeasible point even though the constraints have a feasible point (see Section 1.3).

Sequential quadratic programming methods and interior methods are two alternative approaches to handling the inequality constraints in (1.1). Sequential quadratic programming (SQP) methods find an approximate solution of a sequence of quadratic programming (QP) subproblems in which a quadratic model of the objective function is minimized subject to the linearized constraints. Interior methods approximate a continuous path that passes through a solution of (1.1). In the simplest case, the path is parameterized by a positive scalar parameter μ that may be interpreted as a perturbation for the optimality conditions for the problem (1.1). Both interior methods and SQP methods have an inner/outer iteration structure, with the work for an inner iteration being dominated by the cost of solving a large sparse system of symmetric indefinite linear equations. In the case of SQP methods, these equations involve a subset of the variables and constraints; for interior methods, the equations involve all the constraints and variables.

SQP methods provide a relatively reliable "certificate of infeasibility" and they have the potential of being able to capitalize on a good initial starting point. Sophisticated matrix factorization updating techniques are used to exploit the fact that the linear equations change by only a single row and column at each inner iteration. These updating techniques are often customized for the particular QP method being used and have the benefit of providing a uniform treatment of ill-conditioning and singularity.

On the negative side, it is difficult to implement SQP methods so that exact second derivatives can be used efficiently and reliably. Some of these difficulties stem from the theoretical properties of the quadratic programming subproblem, which can be nonconvex when second derivatives are used. Nonconvex quadratic programming is NP-hard—even for the calculation of a local minimizer [44,73]. The complexity of the QP subproblem has been a major impediment to the formulation of second-derivative SQP methods (although methods based on indefinite QP have been proposed [64, 67]). Over the years, algorithm developers have avoided this difficulty by eschewing second derivatives and by solving a convex QP subproblem defined with a positive semidefinite quasi-Newton approximate Hessian (see, e.g., [85]). There are other difficulties associated with conventional SQP methods that are not specifically related to the use of second derivatives. An SQP algorithm is often tailored to a particular updating technique, e.g., the matrix factors of the Jacobian in the outer iteration can be chosen to match those of the method for the QP subproblem. Any reliance on customized linear algebra software makes it hard to "modernize" a method to reflect new developments in software technology (e.g., in languages that exploit new advances in computer hardware such as multicore processors or GPU-based architectures). Another difficulty is that active-set methods may require a substantial number of QP iterations when the outer iterates are far from the solution. The use of a QP subproblem is motivated by the assumption that the QP objective and constraints provide good “models” of the objective and constraints of the NLP (see Section 2). This should make it unnecessary (and inefficient) to solve the QP to high accuracy during the preliminary iterations. Unfortunately, the simple expedient of limiting the number of inner iterations may have a detrimental effect upon reliability. An approximate QP solution may not predict a sufficient improvement in a merit function (see Section 3.2). Moreover, some of the QP multipliers will have the wrong sign if an active-set method is terminated before a solution is found. This may cause difficulties if the QP multipliers are used to estimate the multipliers for the nonlinear problem. These issues would largely disappear if a primal-dual *interior* method were to be used to solve the QP subproblem. These methods have the benefit of providing a sequence of feasible (i.e., correctly signed) dual iterates. Nevertheless, QP solvers based on conventional interior methods have had limited success within SQP methods because they are difficult to “warm start” from a near-optimal point (see the discussion below). This makes it difficult to capitalize on the property that, as the outer iterates converge, the solution of one QP subproblem is a very good estimate of the solution of the next.

Broadly speaking, the advantages and disadvantages of SQP methods and interior methods complement each other. Interior methods are most efficient when implemented with exact second derivatives. Moreover, they can converge in few inner iterations—even for very large problems. The inner iterates are the iterates of Newton’s method for finding an approximate solution of the perturbed optimality conditions for a given μ. As the dimension and zero/nonzero structure of the Newton equations remains *fixed*, these Newton equations may be solved efficiently using either iterative or direct methods available in the form of advanced “off-the-shelf” linear algebra software. In particular, any new software for multicore and parallel architectures is immediately applicable. Moreover, the perturbation parameter μ plays an auxiliary role as an implicit regularization parameter of the linear equations. This implicit regularization plays a crucial role in the robustness of interior methods on ill-conditioned and ill-posed problems.

On the negative side, although interior methods are very effective for solving “one-off” problems, they are difficult to adapt to solving a sequence of related NLP problems. This difficulty may be explained in terms of the “path-following” interpretation of interior methods. In the neighborhood of an optimal solution, a step *along* the path $x(\mu)$ of perturbed solutions is well-defined, whereas a step *onto* the path from a neighboring point will be extremely sensitive to perturbations in the problem functions (and hence difficult to compute). Another difficulty with conventional interior methods is that a substantial number of iterations may be needed when the constraints are infeasible.

1.1. Notation

Given vectors a and b with the same dimension, the vector with ith component $a_i b_i$ is denoted by $a \cdot b$. The vectors e and e_j denote, respectively, the column vector of ones and the jth column of the identity matrix I. The dimensions of e, e_i and I are defined by the context. Given vectors x and y of dimension n_x and n_y, the $(n_x + n_y)$-vector of elements of x augmented by elements of y is denoted by (x, y). The ith component of a vector labeled with a subscript will be denoted by $(\,\cdot\,)_i$, e.g., $(v_N)_i$ is the ith component of the vector v_N. Similarly, the subvector of components with indices in the index set $\mathcal{S}$ is denoted by $(\,\cdot\,)_{\mathcal{S}}$, e.g., $(v_N)_{\mathcal{S}}$ is the vector with components $(v_N)_i$ for $i \in \mathcal{S}$. The vector with components $\max\{-x_i, 0\}$ (i.e., the magnitude of the negative part of x) is denoted by $[\,x\,]_-$. The vector

p-norm and its subordinate matrix norm is denoted by $\|\cdot\|_p$.

1.2. Background

To simplify the notation, the problem format of (1.1) is modified by introducing slack variables and replacing each general constraint of the form $\ell_i \le \varphi_i(x) \le u_i$ by the equality constraint $\varphi_i(x) - s_i = 0$ and range constraint $\ell_i \le s_i \le u_i$. Without loss of generality, we assume only nonnegativity constraints and use $c(x)$ to denote the vector of combined linear and nonlinear equality constraint functions. (However, we emphasize that the exploitation of the properties of linear constraints is an important issue in the solution of MINLP problems.) The problem to be solved is then

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\;\; f(x) \quad \text{subject to} \quad c(x) = 0, \quad x \ge 0, \tag{1.2}$$

where f and the m components of the constraint vector c are assumed to be twice continuously differentiable for all $x \in \mathbb{R}^n$. Any slack variables are included in the definition of x and c.

Let $g(x)$ denote $\nabla f(x)$, the gradient of f evaluated at x. Similarly, let $J(x)$ denote the $m \times n$ constraint Jacobian with rows formed from the constraint gradients $\nabla c_i(x)$. It is assumed that $J(x)$ has rank m for all x (see the discussion of the use of slack variables in Section 1). Throughout the discussion, the component π_i of the m-vector π will denote the dual variable associated with the constraint $c_i(x) = 0$ or its linearization. Similarly, z_j denotes the dual variable associated with the bound $x_j \ge 0$.

A constraint is *active* at x if it is satisfied with equality. For any *feasible* x, i.e., for any x such that $c(x) = 0$ and $x \ge 0$, all m equality constraints $c_i(x) = 0$ are necessarily active. The indices associated with the active nonnegativity constraints comprise the *active set*, denoted by $\mathcal{A}(x)$, i.e., $\mathcal{A}(x) = \{\, i : x_i = 0 \,\}$. A nonnegativity constraint that is not in the active set is said to be *inactive*. The *inactive set* contains the indices of the inactive constraints, i.e., the so-called "free" variables $\mathcal{I}(x) = \{\, i : x_i > 0 \,\}$.

Under certain constraint regularity assumptions, an optimal solution of (1.2) must satisfy conditions that may be written in terms of the derivatives of the Lagrangian function $L(x,\pi,z) = f(x) - \pi^T c(x) - z^T x$. The triple (x^*,π^*,z^*) is said to be a first-order KKT point for problem (1.2) if it satisfies the KKT conditions

$$\begin{aligned} c(x^*) &= 0, \qquad x^* \ge 0, \\ g(x^*) - J(x^*)^T\pi^* - z^* &= 0, \\ x^* \cdot z^* &= 0, \qquad z^* \ge 0. \end{aligned} \tag{1.3}$$

The property of strict complementarity holds if the vectors x^* and z^* satisfy $x^* \cdot z^* = 0$ with $x^* + z^* > 0$. In keeping with linear programming terminology, we refer to the dual variables π and z as the π*-values* and *reduced costs*, respectively. The vector-triple (x,π,z) is said to constitute a *primal-dual estimate* of the quantities (x^*,π^*,z^*) satisfying (1.3).

The purpose of the constraint regularity assumption is to guarantee that a linearization of the constraints describes the nonlinear constraints with sufficient accuracy that the KKT conditions of (1.3) are necessary for local optimality. One such regularity assumption is the *Mangasarian-Fromovitz constraint qualification* [135, 145], which requires that $J(x^*)$ has rank m, and that there exists a vector p such that $J(x^*)p = 0$ with $p_i > 0$ for all $i \in \mathcal{A}(x^*)$. Another common, but slightly more restrictive, assumption is the *linear independence constraint qualification*, which requires that the matrix of free columns of $J(x^*)$ has full row rank.

Let $H(x,\pi)$ denote the Hessian of $L(x,\pi,z)$ with respect to x, i.e.,

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

Under the linear independence constraint qualification, the second-order necessary optimality conditions require that the first-order conditions (1.3) hold with the additional condition that $p^T H(x^*,\pi^*)p \geq 0$ for all p such that $J(x^*)p = 0$, and $p_i = 0$ for every $i \in \mathcal{A}(x^*)$. See, e.g., Nocedal and Wright [145, Chapter 12] for more discussion of constraint assumptions and optimality conditions.

For a feasible point x, we will denote by $\widehat{J}(x)$ the matrix comprising columns of $J(x)$ corresponding to indices in $\mathcal{I}(x)$. A point x at which $(g(x))_{\mathcal{I}} \in \operatorname{range}(\widehat{J}(x)^T)$ and the linear independence constraint qualification does not hold is said to be *degenerate*. For example, if x is a degenerate vertex, then more than $n - m$ bounds must be active and $\widehat{J}(x)$ has more rows than columns. The Mangasarian-Fromovitz constraint qualification may or may not hold at a degenerate pont. Practical NLP problems with degenerate points are very common and it is crucial that an algorithm be able to handle $\widehat{J}(x)$ with dependent rows. Throughout our discussion of the effects of degeneracy in SQP methods, it will be assumed that the Mangasarian-Fromovitz regularity assumption holds.

1.3. Infeasible problems

In the normal situation, when solving a "one-off" nonlinear program of the form (1.2), one may expect that the problem is feasible—i.e., that there exist points that satisfy the constraints. This is because an infeasible problem is generally the result of a unintended formulation or coding error. However, there are situations when the detection of infeasibility is of particular interest. An example is mixed integer nonlinear programming, where the occurrence of infeasibility is likely as part of a branch and bound fathoming criteria. In this situation, the rapid and reliable detection of infeasibility is a crucial requirement of an algorithm. One way of handling this situation is to define a related *regularized* problem that always has feasible points. This is done by formulating an alternative problem that is always well posed, yet has (x^*,π^*,z^*) as a solution when (x^*,π^*,z^*) exists.

As the question of the existence of a solution revolves around whether or not the constraints admit a feasible point, we can always *relax* the constraints sufficiently to allow the constraints to be feasible. It is then just a question of solving the relaxed problem while simultaneously reducing the amount of relaxation. This process can be automated by introducing *elastic variables* v and w in (1.2), and formulating the *elastic problem*

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n;\; u,v\in\mathbb{R}^m}{\text{minimize}} & f(x) + \rho e^T u + \rho e^T v \\ \text{subject to} & c(x) - u + v = 0, \quad x \geq 0, \quad u \geq 0, \quad v \geq 0, \end{array} \tag{1.4}$$

where ρ is a "penalty" on the elasticity, often referred to as the *elastic weight*, and e is the vector of ones. The smooth elastic problem is equivalent to the nonsmooth bound-constrained problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\;\; f(x) + \rho \sum_{i=1}^{m} |c_i(x)| \quad \text{subject to} \;\; x \geq 0, \tag{1.5}$$

i.e., the elastic problem implicitly enforces a penalty on the sum of the infeasibilities of the constraints $c(x) = 0$. If the original problem is infeasible, then, for large values of ρ, there is a minimizer of the elastic problem that is an $O(1/\rho)$ approximation to a minimizer of the sum of the constraint infeasibilities. This minimizer can be useful in identifying which of the constraints are causing the infeasibility (see Chinneck [35, 36]).

The elastic problem is called an *exact regularization* of (1.2) because if ρ is sufficiently large and (x^*, π^*, z^*) is optimal for (1.2), then it is also optimal for the elastic problem (1.4) with $u = v = 0$. See Fletcher [65, Section 12.3] for a discussion of these issues. The first-order necessary conditions for $(x^*, u^*, v^*, \pi^*, z^*)$ to be an optimal solution for the elastic problem (1.4) are

$$\begin{aligned} c(x^*) - u^* + v^* &= 0, & u^* \ge 0,\;\; v^* \ge 0, && (1.6a)\\ g(x^*) - J(x^*)^T\pi^* - z^* &= 0, & && (1.6b)\\ x^* \cdot z^* &= 0, & z^* \ge 0,\;\; x^* \ge 0 && (1.6c)\\ u^* \cdot (\rho e + \pi^*) = 0,\;\; v^* \cdot (\rho e - \pi^*) &= 0, & -\rho e \le \pi^* \le \rho e. && (1.6d) \end{aligned}$$

To see that the elastic problem (1.4) defines an exact regularization, note that if $\|\pi^*\|_\infty < \rho$, then a solution (x^*, π^*, z^*) of (1.3) is also a solution of (1.6) with $u^* = v^* = 0$. Conditions (1.6) are always necessary for a point (x^*, u^*, v^*) to be an optimal solution for (1.4) because the Mangasarian-Fromovitz constraint qualification is always satisfied.

There are two caveats associated with solving the regularized problem. First, if a solution of the original problem exists, it is generally only a *local* solution of the elastic problem. The elastic problem may be unbounded below, or may have local solutions that are not solutions of the original problem. For example, consider the one-dimensional problem

$$\underset{x\in\mathbb{R}}{\text{minimize}}\;\; x + 1 \quad \text{subject to} \;\; \tfrac{1}{3}x^3 - \tfrac{3}{2}x^2 + 2x = 0, \;\; x \ge 0, \qquad (1.7)$$

which has a unique solution $(x^*, \pi^*) = (0, \frac{1}{2})$. For all $\rho > \frac{1}{2}$, the penalty function (1.5) has a local minimizer $\bar{x} = 2 - O(1/\rho)$ such that $c(\bar{x}) \ne 0$. This example shows that regularization can introduce “phantom” solutions that do not appear in the original problem.

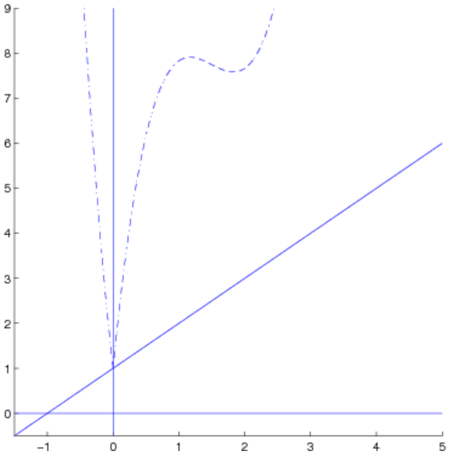

Figure 1: This figure depicts the objective function and penalty function (1.5) for the one-dimensional problem (1.7). The constrained problem has a unique solution $(x^*, \pi^*) = (0, \frac{1}{2})$. However, for all $\rho > \frac{1}{2}$, the penalty function has a local minimizer $\bar{x} = 2 - O(1/\rho)$ with $c(\bar{x}) \ne 0$.

The second caveat is that, in general, the precise definition of the elastic problem is not known in advance because an appropriate value of the parameter ρ depends on the optimal multipliers π^*. This implies that, in practice, any estimate of ρ may need to be increased if the minimization appears to be converging to a regularized solution with $u^* + v^* \neq 0$. If the original problem is infeasible, then $u^* + v^*$ is nonzero for all solutions and ρ must go to infinity if the elastic problem is to provide an estimate of the minimum sum of infeasibilities. Gill, Murray, and Saunders [85] apply an SQP method to a sequence of regularized problems in which ρ is increased geometrically. The sequence is terminated when a solution is found with $u^* + v^* = 0$, or ρ reaches a preassigned upper limit. However, in the context of MINLP, where constraint infeasibility is typical, it is crucial that infeasibility be detected rapidly, which implies that the value of ρ may need to be increased before an accurate solution of the regularized problem is found. (For further information on choosing the value of ρ in this context, see, e.g., Exler and Schittkowski [58], and Byrd, Nocedal, and Waltz [29].)

The objective function in the elastic problem (1.5) is the ℓ_1 penalty function

$$P_1(x;\rho) = f(x) + \rho\|c(x)\|_1 = f(x) + \rho\sum_{i=1}^{m}|c_i(x)|. \tag{1.8}$$

Regularization using an ℓ_1 penalty function is (by far) the most common form of constraint regularization for problems with inequality constraints. However, other exact regularizations can be defined based on using alternative norms to measure the constraint violations. If the ℓ_∞ penalty function

$$P_\infty(x;\rho) = f(x) + \rho\|c(x)\|_\infty = f(x) + \rho\max_{1\le i\le m}|c_i(x)| \tag{1.9}$$

is minimized subject to the constraints $x \ge 0$, an equivalent smooth constrained form of the regularized problem is

$$\underset{x\in\mathbb{R}^n;\ \theta\in\mathbb{R}}{\text{minimize}}\ \ f(x) + \rho\theta \quad \text{subject to}\ \ -\theta e \le c(x) \le \theta e,\ \ x \ge 0,\ \ \theta \ge 0, \tag{1.10}$$

where θ is a temporary nonnegative auxiliary variable. This regularization is exact if $\rho > \|\pi^*\|_1$. The ℓ_2 penalty function $f(x)+\rho\|c(x)\|_2$ also defines an exact regularization, although the use of the two-norm in this form is less common because there is no equivalent smooth constrained form of the problem. (For more on the properties of exact regularization for convex optimization, see Friedlander and Tseng [77].)

The ℓ_2 penalty function is one exception to the rule that a constraint regularization for (1.2) can be written as either a perturbed nonlinearly constrained problem or an equivalent bound-constrained problem, where both formulations depend on the optimal multipliers π^*. However, for some forms of regularization, the dependence on π^* can be explicit (and hence harder to apply). Consider the bound-constrained problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \ f(x) - c(x)^T\pi_E + \tfrac{1}{2}\rho\|c(x)\|_2^2 \quad \text{subject to}\ \ x \ge 0, \tag{1.11}$$

where π_E is an m-vector, and ρ is a nonnegative scalar penalty parameter. Problem (1.11) is used in a number of methods for general nonlinear programming problems based on sequential bound-constrained minimization, see, e.g., [40,41,42,74,76]. The objective function is the well-known Hestenes-Powell augmented Lagrangian, which was first proposed for sequential unconstrained optimization (see, e.g., Hestenes [124], Powell [153], Rockafellar [160], Tapia [171], and Bertsekas [7]). The regularization is exact for $\pi_E = \pi^*$ and all $\rho > \bar\rho$, where $\bar\rho$ depends on the spectral radius of the Hessian of the Lagrangian (and hence, implicitly, on the magnitude of $\|\pi^*\|$). Clearly, this function has a more explicit dependence on π^*. If x_* is a solution of (1.11) for $\pi_E \approx \pi^*$, then x_* is also a solution of the perturbed nonlinearly constrained problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \ f(x) \quad \text{subject to}\ \ c(x) = \mu(\pi_E - \pi_*),\ \ x \ge 0, \tag{1.12}$$

where $\pi_* = \pi_E - \rho c(x_*)$, and μ is the inverse penalty parameter $1/\rho$.

In later sections we consider the close connection between problem regularization and the formulation of SQP methods. In particular, we show that many SQP formulations can be considered in terms of a "plain" SQP method being applied to a related regularized nonlinear program.

Regularization is a very broad idea that can take many forms. For example, a problem formulation may be regularized to ensure that a solution exists or, if there are many solutions, to ensure that a particular favorable solution is found (such as a least-norm solution). Other forms of regularization are specifically designed to make sure that an algorithm may be applied with minimal risk of numerical breakdown. For example, adding slack variables to all constraints, including equalities, guarantees that the Jacobian has full row rank. Such regularization schemes have a beneficial effect on whatever method is used. Some forms of regularization are associated with a specific technique (e.g., trust-region methods impose an implicit regularization on a given subproblem—see Section 3.1.2).

However, although regularization is useful (and sometimes vital) there is usually some price to be paid for its use. In many cases, regularization leads to additional computational overhead or algorithmic complexity. In some cases, regularization will give an approximate rather than exact solution of the problem. More seriously, some forms of regularization lead to the possibility of "phantom" solutions that are not solutions of the original problem.

2. Local Properties of SQP Methods

In many introductory texts, "*the*" SQP method is defined as one in which the quadratic programming subproblem involves the minimization of a quadratic model of the objective function subject to a linearization of the constraints. This description, which broadly defines the original SQP method of Wilson [177] for convex programming, is somewhat over-simplistic for modern SQP methods. Nevertheless, we start by defining a "vanilla" or "plain" SQP method in these terms.

The basic structure of an SQP method involves *inner* and *outer* iterations. Associated with the kth outer iteration is an approximate solution x_k, together with dual variables π_k and z_k for the nonlinear constraints and bounds. Given (x_k, π_k, z_k), new primal-dual estimates are found by solving the quadratic programming subproblem

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f(x_k) + g(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T H(x_k, \pi_k)(x - x_k) \\ \text{subject to} & c(x_k) + J(x_k)(x - x_k) = 0, \quad x \geq 0. \end{array} \tag{2.1}$$

In our plain SQP method, this subproblem is solved by iteration using a quadratic programming method. New estimates π_{k+1} and z_{k+1} of the Lagrange multipliers are the optimal multipliers for the subproblem (2.1). The iterations of the QP method constitute the SQP inner iterations.

The form of the plain QP subproblem (2.1) is motivated by a certain *fixed-point property* that requires the SQP method to terminate in only one (outer) iteration when started at an optimal solution. In particular, the plain QP subproblem is defined in such a way that if $(x_k, \pi_k, z_k) = (x^*, \pi^*, z^*)$, then the NLP primal-dual solution (x^*, π^*, z^*) satisfies the QP optimality conditions for (2.1) and thereby constitutes a solution of the subproblem (see Section 2.2 below for a statement of the QP optimality conditions). Under certain assumptions on the problem derivatives, this fixed-point property implies that $(x_k, \pi_k, z_k) \to (x^*, \pi^*, z^*)$ when the initial point (x_0, π_0, z_0) is sufficiently close to (x^*, π^*, z^*). These assumptions are discussed further below.

Given our earlier statement that SQP methods "minimize a quadratic model of the objective function", readers unfamiliar with SQP methods might wonder why the quadratic term of the quadratic objective of (2.1) involves the Hessian of the Lagrangian function and not the Hessian of the objective function. However, at $(x_k, \pi_k, z_k) = (x^*, \pi^*, z^*)$, the objective of the subproblem defines the second-order local variation of f *on the constraint surface* $c(x) = 0$. Suppose that $x(\alpha)$ is a twice-differentiable feasible path starting at x_k, parameterized by a nonnegative scalar α; i.e., $x(0) = x_k$ and $c(x(\alpha)) = 0$. An inspection of the derivatives $f'(x(\alpha))$ and $f''(x(\alpha))$ at $\alpha = 0$ indicates that the function

$$\widehat{f}_k(x) = f(x_k) + g(x_k)^T\big(x - x_k\big) + \tfrac{1}{2}\big(x - x_k\big)^T H(x_k, \pi_k)\big(x - x_k\big) \tag{2.2}$$

defines a second-order approximation of f for all x lying on $x(\alpha)$, i.e., $\widehat{f}_k(x)$ may be regarded as a *local quadratic model of* f *that incorporates the curvature of the constraints* $c(x) = 0$.

This constrained variation of the objective is equivalent to the *unconstrained* variation of a function known as the *modified Lagrangian*, which is given by

$$L(x; x_k, \pi_k) = f(x) - \pi_k^T(c(x) - \widehat{c}_k(x)), \tag{2.3}$$

where $\widehat{c}_k(x)$ denotes the vector of linearized constraint functions $\widehat{c}_k(x) = c(x_k) + J(x_k)\big(x - x_k\big)$, and $c(x) - \widehat{c}_k(x)$ is known as the *departure from linearity* (see Robinson [158] and Van der Hoek [174]). The first and second derivatives of the modified Lagrangian are given by

$$\begin{aligned}\nabla L(x; x_k, \pi_k) &= g(x) - \big(J(x) - J(x_k)\big)^T \pi_k,\\ \nabla^2 L(x; x_k, \pi_k) &= \nabla^2 f(x) - \sum_{i=1}^{m} (\pi_k)_i \nabla^2 c_i(x).\end{aligned}$$

The Hessian of the modified Lagrangian is independent of x_k and coincides with the Hessian (with respect to x) of the conventional Lagrangian. Also, $L(x; x_k, \pi_k)|_{x=x_k} = f(x_k)$, and $\nabla L(x; x_k, \pi_k)|_{x=x_k} = g(x_k)$, which implies that $\widehat{f}_k(x)$ defines a local quadratic model of $L(x; x_k, \pi_k)$ at $x = x_k$.

Throughout the remaining discussion, g_k, c_k, J_k and H_k denote $g(x)$, $c(x)$, $J(x)$ and $H(x, \pi)$ evaluated at x_k and π_k. With this notation, the quadratic objective is $\widehat{f}_k(x) = f_k + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k)$, with gradient $\widehat{g}_k(x) = g_k + H_k(x - x_k)$. A "hat" will be used to denote quantities associated with the QP subproblem.

2.1. Equality constraints

We motivate some of the later discussion by reviewing the connection between SQP methods and Newton's method for solving a system of nonlinear equations. We begin by omitting the nonnegativity constraints and considering the equality constrained problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) = 0. \tag{2.4}$$

In the case of unconstrained optimization, a standard approach to the formulation of algorithms is to use the first-order optimality conditions to define a system of nonlinear equations $\nabla f(x) = 0$ whose solution is a first-order optimal point x^*. In the constrained case, the relevant nonlinear equations involve the gradient of the Lagrangian function $L(x, \pi)$, which incorporates the first-order feasibility and optimality conditions satisfied by x^* and π^*. If the rows of the constraint Jacobian J at x^* are linearly independent, a primal-dual solution represented by the $n + m$ vector (x^*, π^*) must satisfy the $n + m$ nonlinear equations $F(x, \pi) = 0$, where

$$F(x, \pi) \equiv \nabla L(x, \pi) = \begin{pmatrix} g(x) - J(x)^T \pi \\ -c(x) \end{pmatrix}. \tag{2.5}$$

These equations may be solved efficiently using Newton's method.

2.1.1. Newton's method and SQP

Consider one iteration of Newton's method, starting at estimates x_k and π_k of the primal and dual variables. If v_k denotes the iterate defined by $(n+m)$-vector (x_k, π_k), then the next iterate v_{k+1} is given by

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

Differentiating (2.5) with respect to x and π gives $F'(v) \equiv F'(x,\pi)$ as

$$F'(x,\pi) = \begin{pmatrix} H(x,\pi) & -J(x)^T \\ -J(x) & 0 \end{pmatrix},$$

which implies that the Newton equations may be written as

$$\begin{pmatrix} H_k & -J_k^T \\ -J_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = -\begin{pmatrix} g_k - J_k^T\pi_k \\ -c_k \end{pmatrix},$$

where p_k and q_k denote the Newton steps for the primal and dual variables. If the second block of equations is scaled by -1 we obtain the system

$$\begin{pmatrix} H_k & -J_k^T \\ J_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = -\begin{pmatrix} g_k - J_k^T\pi_k \\ c_k \end{pmatrix}, \tag{2.6}$$

which is an example of a *saddle-point system.* Finally, if the second block of variables is scaled by -1 we obtain an equivalent symmetric system

$$\begin{pmatrix} H_k & J_k^T \\ J_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ -q_k \end{pmatrix} = -\begin{pmatrix} g_k - J_k^T\pi_k \\ c_k \end{pmatrix}, \tag{2.7}$$

which is often referred to as the *KKT system.*

It may not be clear immediately how this method is related to an SQP method. The crucial link follows from the observation that the KKT equations (2.7) represent the first-order optimality conditions for the primal and dual solution (p_k, q_k) of the quadratic program

$$\begin{array}{ll} \underset{p\in\mathbb{R}^n}{\text{minimize}} & (g_k - J_k^T\pi_k)^T p + \tfrac{1}{2}p^T H_k p \\ \text{subject to} & c_k + J_k p = 0, \end{array}$$

which, under certain conditions on the curvature of the Lagrangian discussed below, defines the step from x_k to the point that minimizes the local quadratic model of the objective function subject to the linearized constraints. It is now a simple matter to include the constant objective term f_k (which does not affect the optimal solution) and write the dual variables in terms of $\pi_{k+1} = \pi_k + q_k$ instead of q_k. The equations analogous to (2.7) are then

$$\begin{pmatrix} H_k & J_k^T \\ J_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ -\pi_{k+1} \end{pmatrix} = -\begin{pmatrix} g_k \\ c_k \end{pmatrix}, \tag{2.8}$$

which are the first-order optimality conditions for the quadratic program

$$\underset{p\in\mathbb{R}^n}{\text{minimize}} \;\; f_k + g_k^T p + \tfrac{1}{2}p^T H_k p \quad \text{subject to} \quad c_k + J_k p = 0.$$

When written in terms of the x variables, this quadratic program is

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f_k + g_k^T(x - x_k) + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \\ \text{subject to} & c_k + J_k(x - x_k) = 0. \end{array} \tag{2.9}$$

2.1.2. Local convergence

A standard analysis of Newton's method (see, e.g., Moré and Sorensen [141, Theorem 2.8]) shows that if the KKT matrix is nonsingular at a solution (x^*, π^*), and (x_0, π_0) lies in a sufficiently small neighborhood of (x^*, π^*) in which f and c are twice-continuously differentiable, then the SQP iterates (x_k, π_k) will converge to (x^*, π^*) at a Q-superlinear rate. If, in addition, $H(x, \pi)$ is locally Lipschitz continuous, then the SQP iterates (x_k, π_k) are Q-quadratically convergent. As x is only a subvector of v, with $v = (x, \pi)$, the convergence rate of x_k does not follow immediately. However, as $\|x_k - x^*\| \le \|v_k - v^*\|$, a Q-quadratic rate of convergence of (x_k, π_k) implies an R-quadratic rate of convergence of x_k. For more on the rate of convergence of $\{x_k\}$ relative to $\{x_k, \pi_k\}$, see Ortega and Rheinboldt [148, Chapter 9].

Conditions for the nonsingularity of the KKT matrix may be determined by transforming the KKT system into an equivalent system that reveals the rank. If Q_k is an $n \times n$ nonsingular matrix, then (2.8) is equivalent to the system

$$\begin{pmatrix} Q_k^T H_k Q_k & (J_k Q_k)^T \\ J_k Q_k & 0 \end{pmatrix} \begin{pmatrix} p_Q \\ -\pi_{k+1} \end{pmatrix} = - \begin{pmatrix} Q_k^T g_k \\ c_k \end{pmatrix}, \quad \text{with} \quad p_k = Q_k p_Q. \tag{2.10}$$

Let Q_k be defined so that $J_k Q_k = \begin{pmatrix} 0 & U_k \end{pmatrix}$, where U_k is $m \times m$. The assumption that J_k has rank m implies that U_k is nonsingular. If the n columns of Q_k are partitioned into blocks Z_k and Y_k of dimension $n \times (n-m)$ and $n \times m$, then

$$J_k Q_k = J_k \begin{pmatrix} Z_k & Y_k \end{pmatrix} = \begin{pmatrix} 0 & U_k \end{pmatrix}, \tag{2.11}$$

which shows that $J_k Z_k = 0$ and $J_k Y_k = U_k$. Since Z_k and Y_k are sections of the nonsingular matrix Q_k, they must have independent columns, and, in particular, the columns of Z_k must form a basis for the null-space of J_k. If $Q_k^T H_k Q_k$ and $J_k Q_k$ are partitioned to conform to the Z–Y partition of Q_k, we obtain the block lower-triangular system

$$\begin{pmatrix} U_k & 0 & 0 \\ Z_k^T H_k Y_k & Z_k^T H_k Z_k & 0 \\ Y_k^T H_k Y_k & Y_k^T H_k Z_k & U_k^T \end{pmatrix} \begin{pmatrix} p_Y \\ p_Z \\ -\pi_{k+1} \end{pmatrix} = - \begin{pmatrix} c_k \\ Z_k^T g_k \\ Y_k^T g_k \end{pmatrix}, \tag{2.12}$$

where the $(n-m)$-vector p_Z and m-vector p_Y are the parts of p_Q that conform to the columns of Z_k and Y_k. It follows immediately from (2.12) that the Jacobian $F'(x_k, \pi_k)$ is nonsingular if J_k has independent rows and $Z_k^T H_k Z_k$ is nonsingular. In what follows, we use standard terminology and refer to the vector $Z_k^T g_k$ as the *reduced gradient* and the matrix $Z_k^T H_k Z_k$ as the *reduced Hessian*. If $J(x^*)$ has rank m and the columns of the matrix Z^* form a basis for the null-space of $J(x^*)$, then the conditions: (i) $\nabla L(x^*, \pi^*) = 0$; and (ii) $Z^{*T} H(x^*, \pi^*) Z^*$ positive definite, are sufficient for x^* to be an isolated minimizer of the equality constraint problem (2.4).

2.1.3. Properties of the Newton step

The equations (2.12) have a geometrical interpretation that provides some insight into the properties of the Newton direction. From (2.10), the vectors p_Z and p_Y must satisfy

$$p_k = Q_k p_Q = \begin{pmatrix} Z_k & Y_k \end{pmatrix} \begin{pmatrix} p_Z \\ p_Y \end{pmatrix} = Z_k p_Z + Y_k p_Y.$$

Using block substitution on the system (2.12) we obtain the following equations for p_k and π_{k+1}:

$$\begin{array}{rrcll} & U_k p_Y & = & -c_k, & p_N = Y_k p_Y, \\ & Z_k^T H_k Z_k p_Z & = & -Z_k^T (g_k + H_k p_N), & p_T = Z_k p_Z, \\ p_k = p_N + p_T, & U_k^T \pi_{k+1} & = & Y_k^T (g_k + H_k p_k). & \end{array} \tag{2.13}$$

These equations involve the auxiliary vectors $p_{\scriptscriptstyle N}$ and $p_{\scriptscriptstyle T}$ such that $p_k = p_{\scriptscriptstyle N} + p_{\scriptscriptstyle T}$ and $J_k p_{\scriptscriptstyle T} = 0$. We call $p_{\scriptscriptstyle N}$ and $p_{\scriptscriptstyle T}$ the *normal* and *tangential* steps associated with p_k. Equations (2.13) may be simplified further by introducing the intermediate vector $x_{\scriptscriptstyle F}$ such that $x_{\scriptscriptstyle F} = x_k + p_{\scriptscriptstyle N}$. The definition of the gradient of $\widehat{f}_k$ implies that $g_k + H_k p_{\scriptscriptstyle N} = \nabla \widehat{f}_k(x_k + p_{\scriptscriptstyle N}) = \widehat{g}_k(x_{\scriptscriptstyle F})$, which allows us to rewrite (2.13) in the form

$$\begin{array}{rclrclrcl} & & & U_k p_{\scriptscriptstyle Y} &=& -c_k, & p_{\scriptscriptstyle N} &=& Y_k p_{\scriptscriptstyle Y}, \\ x_{\scriptscriptstyle F} &=& x_k + p_{\scriptscriptstyle N}, & Z_k^T H_k Z_k p_{\scriptscriptstyle Z} &=& -Z_k^T \widehat{g}_k(x_{\scriptscriptstyle F}), & p_{\scriptscriptstyle T} &=& Z_k p_{\scriptscriptstyle Z}, \\ p_k &=& p_{\scriptscriptstyle N} + p_{\scriptscriptstyle T}, & x_{k+1} &=& x_{\scriptscriptstyle F} + p_{\scriptscriptstyle T}, & & & \\ & & & U_k^T \pi_{k+1} &=& Y_k^T \widehat{g}_k(x_{k+1}). & & & \end{array} \tag{2.14}$$

The definition of $x_{\scriptscriptstyle F}$ implies that

$$\widehat{c}_k(x_{\scriptscriptstyle F}) = c_k + J_k(x_{\scriptscriptstyle F} - x_k) = c_k + J_k p_{\scriptscriptstyle N} = c_k + J_k Y_k p_{\scriptscriptstyle Y} = c_k + U_k p_{\scriptscriptstyle Y} = 0,$$

which implies that the normal component $p_{\scriptscriptstyle N}$ satisfies $J_k p_{\scriptscriptstyle N} = -c_k$ and constitutes the Newton step from x_k to the point $x_{\scriptscriptstyle F}$ satisfying the linearized constraints $c_k + J_k(x - x_k) = 0$. On the other hand, the tangential step $p_{\scriptscriptstyle T}$ satisfies $p_{\scriptscriptstyle T} = Z_k p_{\scriptscriptstyle Z}$, where $Z_k^T H_k Z_k p_{\scriptscriptstyle Z} = -Z_k^T \widehat{g}_k(x_{\scriptscriptstyle F})$. If the reduced Hessian $Z_k^T H_k Z_k$ is positive definite, which will be the case if x_k is sufficiently close to a locally unique (i.e., isolated) minimizer of (2.4), then $p_{\scriptscriptstyle T}$ defines the Newton step from $x_{\scriptscriptstyle F}$ to the *minimizer* of the quadratic model $\widehat{f}_k(x)$ in the subspace orthogonal to the constraint normals (i.e., on the surface of the linearized constraint $\widehat{c}_k(x) = 0$). It follows that the Newton direction is the sum of two steps: a normal step to the linearized constraint and the tangential step on the constraint surface that minimizes the quadratic model. This property reflects the two (usually conflicting) underlying processes present in all algorithms for optimization—the minimization of the objective and the satisfaction of the constraints.

In the discussion above, the normal step $p_{\scriptscriptstyle N}$ is interpreted as a Newton direction for the equations $\widehat{c}_k(x) = 0$ at $x = x_k$. However, in some situations, $p_{\scriptscriptstyle N}$ may also be interpreted as the solution of a minimization problem. The Newton direction p_k is unique, but the decomposition $p_k = p_{\scriptscriptstyle T} + p_{\scriptscriptstyle N}$ depends on the choice of the matrix Q_k associated with the Jacobian factorization (2.11). If Q_k is orthogonal, i.e., if $Q_k^T Q_k = I$, then $Z_k^T Y_k = 0$ and the columns of Y_k form a basis for the range space of J_k^T. In this case, $p_{\scriptscriptstyle N}$ and $p_{\scriptscriptstyle T}$ define the unique range-space and null-space decomposition of p_k, and $p_{\scriptscriptstyle N}$ is the unique solution with least two-norm of the least-squares problem

$$\min_p \; \|\widehat{c}_k(x_k) + J_k p\|_2, \quad \text{or, equivalently,} \quad \min_p \; \|c_k + J_k p\|_2.$$

This interpretation is useful in the formulation of variants of Newton's method that do not require (x_k, π_k) to lie in a small neighborhood of (x^*, π^*). In particular, it suggests a way of computing the normal step when the equations $J_k p = -c_k$ are not compatible.

For consistency with the inequality constrained case below, the primal-dual solution of the kth QP subproblem is denoted by $(\widehat{x}_k, \widehat{\pi}_k)$. With this notation, the first-order optimality conditions for the QP subproblem (2.9) are given by

$$\begin{aligned} J_k(\widehat{x}_k - x_k) + c_k &= 0, \\ g_k + H_k(\widehat{x}_k - x_k) - J_k^T \widehat{\pi}_k &= 0. \end{aligned} \tag{2.15}$$

Similarly, the Newton iterates are given by $x_{k+1} = \widehat{x}_k = x_k + p_k$ and $\pi_{k+1} = \widehat{\pi}_k = \pi_k + q_k$.

2.1.4. Calculation of the Newton step

There are two broad approaches for solving the Newton equations (either in saddle-point form (2.6) or symmetric form (2.7). The first involves solving the full $n+m$ KKT equations, the second decomposes the KKT equations into the three systems associated with the block lower-triangular equations (2.12).

In the full-matrix approach, the matrix K may be represented by its *symmetric indefinite factorization* (see, e.g., Bunch and Parlett [21], and Bunch and Kaufman [19]):

$$PKP^T = LDL^T, \tag{2.16}$$

where P is a permutation matrix, L is lower triangular and D is block diagonal, with 1×1 or 2×2 blocks. (The latter are required to retain numerical stability.) Some prominent software packages include MA27 (Duff and Reid [56]), MA57 (Duff [55]), MUMPS (Amestoy et al. [2]), PARDISO (Schenk and Gärtner [164]), and SPOOLES (Ashcraft and Grimes [5]).

The decomposition approach is based on using an explicit or implicit representation of the null-space basis matrix Z_k. When J_k is dense, Z_k is usually computed directly from a QR factorization of J_k (see, e.g., Coleman and Sorensen [39], and Gill et al. [87]). When J_k is sparse, however, known techniques for obtaining an orthogonal *and sparse* Z may be expensive in time and storage, although some effective algorithms have been proposed (see, e.g., Coleman and Pothen [38]; Gilbert and Heath [79]).

The representation of Z_k most commonly used in sparse problems is called the *variable-reduction* form of Z_k, and is obtained as follows. The columns of J_k are partitioned so as to identify explicitly an $m \times m$ nonsingular matrix B (the *basis matrix*). Assuming that B is at the "left" of J_k, we have

$$J_k = \begin{pmatrix} B & S \end{pmatrix}.$$

(In practice, the columns of B may occur anywhere.) When J_k has this form, a basis for the null space of J_k is given by the columns of the (non-orthogonal) matrix Q_k defined as

$$Q_k = \begin{pmatrix} -B^{-1}S & I_m \\ I_{n-m} & \end{pmatrix}, \quad \text{with} \quad Z_k = \begin{pmatrix} -B^{-1}S \\ I_{n-m} \end{pmatrix} \quad \text{and} \quad Y_k = \begin{pmatrix} I_m \\ 0 \end{pmatrix}.$$

This definition of Q_k means that matrix-vector products $Z_k^T v$ or $Z_k v$ can be computed using a factorization of B (typically, a sparse LU factorization; see Gill, Murray, Saunders and Wright [92]), and Z_k need not be stored explicitly.

For large sparse problems, the reduced Hessian $Z_k^T H_k Z_k$ associated with the solution of (2.14) will generally be much more dense than H_k and B. However, in many cases, $n - m$ is small enough to allow the storage of a dense Cholesky factor of $Z_k^T H_k Z_k$.

2.2. Inequality constraints

Given an approximate primal-dual solution (x_k, π_k) with $x_k \geq 0$, an outer iteration of a typical SQP method involves solving the QP subproblem (2.1), repeated here for convenience:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & f_k + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k (x - x_k) \\ \text{subject to} & J_k(x - x_k) = -c_k, \quad x \geq 0. \end{array} \tag{2.17}$$

Assume for the moment that this QP subproblem is feasible, with primal-dual solution $(\widehat{x}_k, \widehat{\pi}_k, \widehat{z}_k)$. The next plain SQP iterate is $x_{k+1} = \widehat{x}_k$, $\pi_{k+1} = \widehat{\pi}_k$ and $z_{k+1} = \widehat{z}_k$. The QP first-order optimality conditions are

$$\begin{aligned} J_k(\widehat{x}_k - x_k) + c_k &= 0, & \widehat{x}_k &\geq 0; \\ g_k + H_k(\widehat{x}_k - x_k) - J_k^T \widehat{\pi}_k - \widehat{z}_k &= 0, & & \\ \widehat{x}_k \cdot \widehat{z}_k &= 0, & \widehat{z}_k &\geq 0. \end{aligned} \tag{2.18}$$

Let $p_k = \widehat{x}_k - x_k$ and let $\bar{p}_k$ denote the vector of free components of p_k, i.e., the components with indices in $\mathcal{I}(\widehat{x}_k)$. Similarly, let $\bar{z}_k$ denote the free components of $\widehat{z}_k$. The complementarity conditions imply that $\bar{z}_k = 0$ and we may combine the first two sets of equalities in (2.18) to give

$$\begin{pmatrix} \bar{H}_k & \bar{J}_k^T \\ \bar{J}_k & 0 \end{pmatrix} \begin{pmatrix} \bar{p}_k \\ -\widehat{\pi}_k \end{pmatrix} = - \begin{pmatrix} (g_k + H_k \eta_k)_{\mathcal{I}} \\ c_k + J_k \eta_k \end{pmatrix}, \tag{2.19}$$

where $\bar{J}_k$ is the matrix of free columns of J_k, and η_k is the vector

$$(\eta_k)_i = \begin{cases} (\widehat{x}_k - x_k)_i & \text{if } i \in \mathcal{A}(\widehat{x}_k); \\ 0 & \text{if } i \in \mathcal{I}(\widehat{x}_k). \end{cases}$$

If the active sets at $\widehat{x}_k$ and x_k are the same, i.e., $\mathcal{A}(\widehat{x}_k) = \mathcal{A}(x_k)$, then $\eta_k = 0$. If $\widehat{x}_k$ lies in a sufficiently small neighborhood of a *nondegenerate* solution x^*, then $\mathcal{A}(\widehat{x}_k) = \mathcal{A}(x^*)$ and hence $\bar{J}_k$ has full row rank (see Robinson [159]). In this case we say that the QP *identifies the correct active set* at x^*. If, in addition, (x^*, π^*) satisfies the second-order sufficient conditions for optimality, then KKT system (2.19) is nonsingular and the plain SQP method is equivalent to Newton's method applied to the equality-constraint subproblem defined by fixing the variables in the active set at their bounds.

However, at a *degenerate* QP solution, the rows of $\bar{J}_k$ are linearly dependent and the KKT equations (2.19) are compatible but singular. Broadly speaking, there are two approaches to dealing with the degenerate case, where each approach is linked to the method used to solve the QP subproblem. The first approach employs a QP method that not only finds the QP solution $\widehat{x}_k$, but also identifies a "basic set" of variables that define a matrix $\widetilde{J}_k$ with *linearly independent* rows. The second approach solves a *regularized* or *perturbed* QP subproblem that provides a perturbed version of the KKT system (2.19) that is nonsingular for any $\bar{J}_k$.

Identifying independent constraints. The first approach is based on using a QP algorithm that provides a primal-dual QP solution that satisfies a *nonsingular* KKT system analogous to (2.19). A class of quadratic programming methods with this property are primal-feasible active-set methods, which form the basis of the software packages NPSOL and SNOPT. Primal-feasible QP methods have two phases: in phase 1, a feasible point is found by minimizing the sum of infeasibilities; in phase 2, the quadratic objective function is minimized while feasibility is maintained. In each iteration, the variables are labeled as being "basic" or "nonbasic", where the nonbasic variables are temporarily fixed at their current value. The indices of the basic and nonbasic variables are denoted by $\mathcal{B}$ and $\mathcal{N}$ respectively. A defining property of the $\mathcal{B}$–$\mathcal{N}$ partition is that the rows of the Jacobian appearing in the KKT matrix are always linearly independent. Once an initial basic set is identified, all subsequent KKT equations have a constraint block with independent rows. (For more details of primal-feasible active-set methods, see Section A.1 of the Appendix.)

Let $p_k = \widehat{x}_k - x_k$, where $(\widehat{x}_k, \widehat{\pi}_k)$ is the QP solution found by a primal-feasible active-set method. Let $\widetilde{p}_k$ denote the vector of components of p_k in the final basic set $\mathcal{B}$, with $\widetilde{J}_k$ the corresponding columns of J_k. The vector $(\widetilde{p}_k, \widehat{\pi}_k)$ satisfies the *nonsingular* KKT equations

$$\begin{pmatrix} \widetilde{H}_k & \widetilde{J}_k^T \\ \widetilde{J}_k & 0 \end{pmatrix} \begin{pmatrix} \widetilde{p}_k \\ -\widehat{\pi}_k \end{pmatrix} = - \begin{pmatrix} (g_k + H_k \eta_k)_{\mathcal{B}} \\ c_k + J_k \eta_k \end{pmatrix}, \tag{2.20}$$

where η_k is now defined in terms of the final QP nonbasic set, i.e.,

$$(\eta_k)_i = \begin{cases} (\widehat{x}_k - x_k)_i & \text{if } i \in \mathcal{N}; \\ 0 & \text{if } i \notin \mathcal{N}. \end{cases} \tag{2.21}$$

As in (2.19), if the basic-nonbasic partition is not changed during the solution of the subproblem, then $\eta_k = 0$. If this final QP nonbasic set is used to define the initial nonbasic set for the next QP subproblem, it is typical for the later QP subproblems to reach optimality in a *single iteration* because the solution of the first QP KKT system satisfies the QP optimality conditions immediately. In this case, the phase-1 procedure simply performs a feasibility check that would be required in any case.

Constraint regularization. One of the purposes of regularization is to define KKT equations that are nonsingular regardless of the rank of $\bar{J}_k$. Consider the perturbed version of equations (2.19) such that

$$\begin{pmatrix} \bar{H}_k & \bar{J}_k^T \\ \bar{J}_k & -\mu I \end{pmatrix} \begin{pmatrix} \bar{p}_k \\ -\widehat{\pi}_k \end{pmatrix} = - \begin{pmatrix} (g_k + H_k \eta_k)_{\mathcal{I}} \\ c_k + J_k \eta_k \end{pmatrix}, \tag{2.22}$$

where μ is a small positive constant. In addition, assume that $\bar{Z}_k^T \bar{H}_k \bar{Z}_k$ is positive definite, where the columns of $\bar{Z}_k$ form a basis for the null space of $\bar{J}_k$. With this assumption, the unperturbed KKT equations (2.19) are singular if and only if $\bar{J}_k$ has linearly dependent rows.

For simplicity, assume that $\eta_k = 0$. Let $(\,U \;\; V\,)$ be an orthonormal matrix such that the columns of U form a basis for $\text{null}(\bar{J}_k^T)$ and the columns of V form a basis for $\text{range}(\bar{J}_k)$. The unique expansion $\widehat{\pi}_k = U\pi_U + V\pi_V$ allows us to rewrite (2.22) as

$$\begin{pmatrix} \bar{H}_k & \bar{J}_k^T V & \\ V^T \bar{J}_k & -\mu I & \\ & & -\mu I \end{pmatrix} \begin{pmatrix} \bar{p}_k \\ -\pi_V \\ -\pi_U \end{pmatrix} = - \begin{pmatrix} (g_k)_{\mathcal{I}} \\ V^T c_k \\ 0 \end{pmatrix}, \tag{2.23}$$

where $\bar{J}_k^T U = 0$ from the definition of U, and $U^T c_k = 0$ because $c_k \in \text{range}(\bar{J}_k)$. The following simple argument shows that the equations (2.23) are nonsingular, regardless of the rank of $\bar{J}_k$. First, observe that $V^T \bar{J}_k$ has full row rank. Otherwise, if $v^T V^T \bar{J}_k = 0$, it must be the case that $Vv \in \text{null}(\bar{J}_k^T)$. But since $Vv \in \text{range}(V)$ and $\text{range}(V)$ is orthogonal to $\text{null}(\bar{J}_k^T)$, we conclude that $Vv = 0$, and the linearly independence of the columns of V gives $v = 0$.

Moreover, equations (2.23) imply that $\pi_U = 0$ and $\widehat{\pi}_k \in \text{range}(\bar{J}_k)$. If $(g_{k+1})_{\mathcal{I}}$ denotes the free components of $g_{k+1} = g_k + Hp_k$, then

$$\bar{J}_k^T \widehat{\pi}_k = (g_{k+1})_{\mathcal{I}} \quad \text{and} \quad \widehat{\pi}_k \in \text{range}(\bar{J}_k).$$

These are the necessary and sufficient conditions for $\widehat{\pi}_k$ to be the unique least-length solution of the compatible equations $\bar{J}_k^T \pi = (g_{k+1})_{\mathcal{I}}$. This implies that the regularization gives a unique vector of multipliers.

Wright [178, 179, 180] and Hager [119] show that an SQP method using the regularized equations (2.22) will converge at a superlinear rate, even in the degenerate case. In Section A.3 of the Appendix, QP methods are discussed that give equations of the form (2.22) at every outer iteration, not just in the neighborhood of the solution. These methods implicitly shift the constraints by an amount of order μ and give QP multipliers that converge to an $O(\mu)$ estimate of the least-length multipliers.

A related regularization scheme has been proposed and analyzed by Fischer [59], who solves a second QP to obtain the multiplier estimates. Anitescu [4] regularizes the problem by imposing a trust-region constraint on the plain SQP subproblem (2.1) and solving the resulting subproblem by a semidefinite programming method.

3. The formulation of modern SQP methods

SQP methods have evolved considerably since Wilson's thesis appeared in 1963. Current implementations of SQP methods for large-scale optimization have solved problems with as many as 40,000 variables and inequality constraints (see, e.g., Gill, Murray and Saunders [85]). During this evolution, both the theory and practice of SQP methods have benefited substantially from developments in competing methods. Similarly, research in SQP methods has had a considerable impact on the formulation and analysis of rival methods—for example, on the treatment of equality constraints in interior methods. On the surface, many recent SQP methods bear little resemblance to the plain SQP method proposed by Wilson. In this section we review some of the principal developments in SQP methods since 1963 while emphasizing connections to other methods. In our discussion, we use the broad definition of an SQP method as one that uses a quadratic programming subproblem to estimate the active set. Implicit in this definition is the assumption that, in the neighborhood of the solution, an SQP method will solve the Newton KKT equations (or some approximation) defined in terms of the free variables.

The complex interrelationships that exist between optimization methods make it difficult (and controversial) to give a precise taxonomy of the many different SQP approaches. Instead, we will discuss methods under four topics that, in our opinion, were influential in shaping developments in the area. Each of these topics will provide a starting-point for discussion of related methods and extensions. The topics are: (i) merit functions and the Han-Powell SQP method, (ii) sequential unconstrained methods, (iii) line-search and trust-region filter methods, and (iv) methods that solve a convex program to determine an estimate of the active set. The modern era of SQP methods can be traced to the publication of the Han-Powell method in 1976 [120, 155]. (It may be argued that almost all subsequent developments in SQP methods are based on attempts to correct perceived theoretical and practical deficiencies in the Wilson-Han-Powell approach.) The sequential unconstrained approaches to SQP have evolved from a 1982 paper by Fletcher [62, 63]. Filter SQP methods are a more recent development, being proposed by Fletcher and Leyffer [67, 68] in 1998.

3.1. Review of line-search and trust-region methods

Our discussion of the equality constrained problem in Section 2.1.1 emphasizes the local equivalence between a plain SQP method and Newton's method applied to the first-order optimality conditions. As the Newton iterates may diverge or may not be well-defined if the starting point is not sufficiently close to a solution, some modification is needed to force convergence from arbitrary starting points. Line-search methods and trust-region methods are two alternative modifications of Newton's method. We begin by reviewing the main properties of these methods in the context of unconstrained minimization.

3.1.1. Line-search methods: the unconstrained case

Associated with the kth iteration of a conventional line-search method for unconstrained optimization is a scalar-valued function $m_k(x)$ that represents a *local line-search model* of f. The next iterate is then $x_{k+1} = x_k + d_k$, where d_k is chosen so that the improvement in f is at least as good as a fixed fraction of the improvement in the local model, i.e., d_k must satisfy

$$f(x_k) - f(x_k + d_k) \ge \eta\big(m_k(x_k) - m_k(x_k + d_k)\big), \tag{3.1}$$

where η is a fixed parameter such that $0 < \eta < \frac{1}{2}$. Typical line-search models are affine and quadratic functions based on a first- or second-order Taylor-series approximation of f. For example, a first-order approximation provides the affine line-search model $m_k(x) = f(x_k) + g(x_k)^T(x - x_k)$. In a general line-search method, the change in variables has the form $d_k \equiv d_k(\alpha_k)$, where α_k is a scalar steplength that defines a point on the parameterized path $d_k(\alpha)$. In the simplest case, $d_k(\alpha) = \alpha p_k$, where p_k is an approximate solution of the unconstrained subproblem $\min_{p\in\mathbb{R}^n} g_k^Tp + \frac{1}{2}p^TB_kp$, with B_k a *positive-definite* approximation of the Hessian H_k. More generally, if H_k is indefinite, $d_k(\alpha)$ is defined in terms of p_k and a direction s_k such that $s_k^TH_ks_k < 0$ (see, e.g., Goldfarb [101], Moré and Sorensen [140], and Olivares, Moguerza and Prieto [146]). A crucial feature of a line-search method is that $d_k(\alpha)$ is defined in terms of a *convex* subproblem, which may be defined implicitly during the calculation of p_k; see, e.g., Greenstadt [114], Gill and Murray [82], Schnabel and Eskow [168]).

Condition (3.1) may be written as $f(x_k) - f(x_k + d_k) \geq \eta\Delta m_k(d_k)$, where the quantity

$$\Delta m_k(d) = m_k(x_k) - m_k(x_k + d) \tag{3.2}$$

is the change in f predicted by the line-search model function. An essential property of the line-search model is that it must always be possible to find an α_k that satisfies (3.1). In particular, there must exist a positive $\bar{\alpha}$ such that

$$f(x_k + d_k(\alpha)) \leq f(x_k) - \eta\Delta m_k\big(d_k(\alpha)\big), \quad \text{for all } \alpha \in (0, \bar{\alpha}). \tag{3.3}$$

For this condition to hold, the model must predict a reduction in $f(x)$ at $x = x_k$, i.e., $\Delta m_k\big(d_k(\alpha)\big) > 0$ for all $\alpha \in (0, \bar{\alpha})$. Under the assumption that (3.3) holds, there are various algorithms for finding an appropriate α_k. For example, in a *backtracking line search*, the step $\alpha_k = 1$ is decreased by a fixed factor until condition (3.1) is satisfied. It can be shown that this simple procedure is enough to guarantee a sufficient decrease in f. More sophisticated methods satisfy (3.1) in conjunction with other conditions that ensure a sufficient decrease in f (see, e.g., Ortega and Rheinboldt [147], Moré and Thuente [142], and Gill et al. [88]).

The line-search methods defined above enforce a monotone decrease in f at each iteration. In some cases the definition of $d_k(\alpha)$ may warrant the use of a *nonmonotone* line search in which f is permitted to increase on some iterations. An example of a nonmonotone line-search condition is

$$f(x_k + d_k(\alpha)) \leq \max_{0\leq j\leq r} \,[f(x_{k-j})] - \eta\Delta m_k\big(d_k(\alpha)\big),$$

where r is some fixed number of previous iterations (for other schemes of varying complexity, see, e.g., Grippo, Lampariello and Lucidi [115, 116, 117], Toint [172], and Zhang and Hager [185]). In Section 3.2, we discuss the “watchdog technique”, which is a nonmonotone line search that allows the value $\alpha_k = 1$ to be used for a limited number of steps, regardless of the value of f.

3.1.2. Trust-region methods: the unconstrained case

When there are no constraints, line-search methods and trust-region methods have many properties in common. Both methods choose the value of a scalar variable so that the objective improves by an amount that is at least as good as a fraction of the improvement in a local model (see condition (3.1)). A crucial difference is that a line-search method involves the solution of a bounded *convex* subproblem. By contrast, trust-region methods solve a constrained, possibly nonconvex, subproblem of the form

$$\min_{d\in\mathbb{R}^n} \; g_k^Td + \tfrac{1}{2}d^TH_kd \quad \text{subject to} \quad \|d\| \leq \delta_k, \tag{3.4}$$

with condition (3.1) being enforced, if necessary, by reducing the positive scalar δ_k (the trust-region radius). The final value of δ_k is also used to define an initial estimate of δ_{k+1}, with the possibility that δ_{k+1} is increased to a multiple of δ_k if the reduction in f is significantly better than the reduction predicted by the model. If the trust-region radius is reduced over a sequence of consecutive iterations, the step d_k will go to zero along the direction of steepest descent with respect to the particular norm used to define the trust region. As in a line search, it is possible to define trust-region methods that do not enforce a reduction in f at every step (see, e.g., Gu and Mo [118]).

The complexity of constrained minimization is generally higher than that of unconstrained minimization. Moreover, the trust-region subproblem may need to be solved more than once before the condition (3.1) is satisfied. Nevertheless, trust-region methods provide computational benefits when some of the eigenvalues of H_k are close to zero (see Kroyan [128]). Modern trust-region methods require only an approximate solution of (3.4). For a comprehensive review of trust-region methods for both unconstrained and constrained optimization, see Conn, Gould and Toint [43].

3.2. The Han-Powell method

Han [121] and Powell [155] introduced two crucial improvements to the plain SQP method of Wilson. The first was the use of a QP subproblem defined in terms of a *positive-definite* quasi-Newton approximation. The second was the use of a *line-search merit function* to obtain a sequence of improving estimates of the solution.

A merit function $\mathcal{M}$ is a scalar-valued function whose value provides a measure of the quality of a given point as an estimate of a solution of the constrained problem. Each value of $\mathcal{M}$ represents a compromise between the (usually conflicting) aims of minimizing the objective function and minimizing the constraint violations. Analogous to the unconstrained case, the merit function is used in conjunction with a line-search model $m_k(x)$ to define a sufficient decrease at the kth iteration. In the constrained case, d_k is chosen to satisfy

$$\mathcal{M}(x_k) - \mathcal{M}(x_k + d_k) \ge \eta\big(m_k(x_k) - m_k(x_k + d_k)\big), \quad x_k + d_k \ge 0. \tag{3.5}$$

Han and Powell proposed the use of the ℓ_1 penalty function (1.8) as a merit function, i.e., $\mathcal{M}(x) \triangleq \mathcal{M}(x;\rho) = P_1(x;\rho)$. Moreover, they suggested that $d_k(\alpha) = \alpha p_k = \alpha(\widehat{x}_k - x_k)$, where $\widehat{x}_k$ is the solution of the convex subproblem

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f_k + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k) \\ \text{subject to} & c_k + J_k(x - x_k) = 0, \quad x \ge 0, \end{array} \tag{3.6}$$

with B_k a positive-definite approximation to the Hessian of the Lagrangian or augmented Lagrangian (in Section 3.2.1 below, we discuss the definition of B_k in this context). As the QP (3.6) is convex, it may be solved using either a primal or dual active-set method (see Section A.1 of the Appendix). In either case, the QP multipliers $\widehat{\pi}_k$, and vector $\widetilde{p}_k$ of components of p_k in the final basic set satisfy the nonsingular KKT equation

$$\begin{pmatrix} \widetilde{B}_k & \widetilde{J}_k^T \\ \widetilde{J}_k & \end{pmatrix} \begin{pmatrix} \widetilde{p}_k \\ -\widehat{\pi}_k \end{pmatrix} = -\begin{pmatrix} (g_k + B_k\eta_k)_{\mathcal{B}} \\ c_k + J_k\eta_k \end{pmatrix}, \tag{3.7}$$

where $\widetilde{B}_k$ and $\widetilde{J}_k$ denote the matrices of basic components of B_k and J_k and η_k is defined as in (cf. (2.21)).

3.2.1. Quasi-Newton approximations

Many methods for unconstrained minimization use a quasi-Newton approximation of the Hessian when second derivatives are either unavailable or too expensive to evaluate. Arguably, the most commonly used quasi-Newton approximation is defined using the BFGS method (see Broyden [15], Fletcher [60], Goldfarb [100], and Shanno [169]). Given iterates x_k and x_{k+1}, and a symmetric approximate Hessian B_k, the BFGS approximation for the next iteration has the form

$$B_{k+1} = B_k - \frac{1}{d_k^T B_k d_k} B_k d_k d_k^T B_k + \frac{1}{y_k^T d_k} y_k y_k^T, \tag{3.8}$$

where $d_k = x_{k+1} - x_k$ and $y_k = g(x_{k+1}) - g(x_k)$. If B_k is positive definite, then B_{k+1} is positive definite if and only if the approximate curvature $y_k^T d_k$ is positive.

For the constrained case, Han [121] proposed maintaining a BFGS approximation of the Hessian of the augmented Lagrangian function

$$L_A(x, \pi; \rho) = f(x) - c(x)^T \pi + \tfrac{1}{2} \rho c(x)^T c(x).$$

(As the Hessian of the Lagrangian does not include the linear constraints, we have omitted z from the Lagrangian term.) This implies that the gradient difference y_k in (3.8) involves the gradient of the augmented Lagrangian, with

$$d_k = x_{k+1} - x_k, \quad \text{and} \quad y_k = \nabla_x L_A(x_{k+1}, \pi_{k+1}; \rho) - \nabla_x L_A(x_k, \pi_{k+1}; \rho),$$

where π_{k+1} are estimates of the optimal dual variables. This proposal is motivated by the fact that if ρ is sufficiently large, the Hessian of $L(x, \pi; \rho)$ is positive definite for all (x, π) close to an isolated solution (x^*, π^*) (see also, Tapia [170], and Byrd, Tapia and Zhang [30]).

The use of an augmented Lagrangian Hessian for the QP subproblem changes the properties of the QP dual variables. In particular, if $(\widehat{x}_k, \widehat{\pi}_k, \widehat{z}_k)$ is the solution of the QP (3.6) with B_k defined as $H_k + \rho J_k^T J_k$, then $(\widehat{x}_k, \widehat{\pi}_k + \rho c_k, \widehat{z}_k)$ is the solution of the QP (3.6) with B_k replaced by H_k (assuming that the same local solution is found when H_k is not positive definite). In other words, if the augmented Lagrangian Hessian is used instead of the Lagrangian Hessian, the x and z variables do not change, but the π-values are shifted by ρc_k. An appropriate value for π_{k+1} in the definition of y_k is then $\pi_{k+1} = \widehat{\pi}_k + \rho c_k$, giving, after some simplification,

$$y_k = g_{k+1} - g_k - (J_{k+1} - J_k)^T \widehat{\pi}_k + \rho J_{k+1}^T (c_{k+1} - c_k).$$

If the approximate curvature $y_k^T d_k$ is not positive, the matrix B_{k+1} of (3.8) is either indefinite or undefined. In terms of an update to the Hessian of the augmented Lagrangian, a negative $y_k^T d_k$ implies that either ρ is not sufficiently large, or the curvature of the penalty term $\frac{1}{2}\rho c(x)^T c(x)$ is negative along d_k. In the first case, ρ must be increased by an amount that is sufficiently large to give a positive value of $y_k^T d_k$. In the second case, the approximate curvature of the Lagrangian is not sufficiently positive and there is no finite ρ that gives $y_k^T d_k > 0$. In this case, the update should be skipped. The curvature is considered not sufficiently positive if

$$y_k^T d_k < \sigma_k, \qquad \sigma_k = \alpha_k (1 - \eta) p_k^T B_k p_k, \tag{3.9}$$

where η is a preassigned constant $(0 < \eta < 1)$ and p_k is the search direction $\widehat{x}_k - x_k$ defined by the QP subproblem. If $y_k^T d_k < \sigma_k$, then ρ is replaced by $\rho + \Delta\rho$, where

$$\Delta\rho = \begin{cases} \dfrac{\sigma_k - y_k^T d_k}{d_k^T J_{k+1}^T (c_{k+1} - c_k)}, & \text{if } d_k^T J_{k+1}^T (c_{k+1} - c_k) > 0; \\ 0, & \text{otherwise.} \end{cases}$$

If $\Delta\rho = 0$, the approximate curvature of $c(x)^T c(x)$ is not positive and the update should be skipped.

Maintaining an approximation of the Hessian of $L_A(x;\pi,\rho)$ involves a number of difficulties, all of which stem from the need to increase the value of ρ. First, the usual convergence of the sequence $\{B_k\}$ is disrupted when ρ is increased. Second, a large increase in ρ will give an ill-conditioned matrix B_{k+1}. Finally, because ρ is always increased, the ill-effects of large values of ρ persist throughout the computation.

Powell [155] suggested the use of a positive-definite BFGS approximation for the *Lagrangian* Hessian, i.e., the update pair is

$$d_k = x_{k+1} - x_k, \quad y_k = \nabla_{\!x} L(x_{k+1}, \pi_{k+1}, z_{k+1}) - \nabla_{\!x} L(x_k, \pi_{k+1}, z_{k+1}). \tag{3.10}$$

If the QP multipliers are used for π_{k+1}, the difference in Lagrangian gradients is given by $y_k = g_{k+1} - g_k - (J_{k+1} - J_k)^T \widehat{\pi}_k$.

A *positive-definite* BFGS approximation may appear to be a surprising choice for B_k, given that the Hessian of the Lagrangian is generally *indefinite* at the solution. However, Powell's proposal is based on the observation that the approximate curvature is likely to be positive in the neighborhood of an isolated solution, even when the Hessian of the Lagrangian is indefinite. The reason for this is that the iterates of a quasi-Newton SQP converge to the solution along a path that lies in the null space of the "free" columns of the Jacobian. As the Lagrangian Hessian is generally positive definite along this path, the approximate curvature $y_k^T d_k$ is positive as the iterates converge and an R-superlinear convergence rate is obtained. Powell's proposal may be justified by considering the properties of $(\widehat{x}_k, \widehat{\pi}_k)$, the solution of the QP subproblem. Let $p_k = \widehat{x}_k - x_k$ and $\widehat{g}(x) = g_k + B_k(x - x_k)$. It is shown in Section A.4.1 of the Appendix that $(\widehat{x}_k, \widehat{\pi}_k)$ satisfies the equations

$$\begin{array}{rclrclrcl} & & & U_k p_Y &=& -c_k, & p_N &=& Y_k p_Y, \\ x_F &=& x_k + p_N, & Z_k^T B_k Z_k p_Z &=& -Z_k^T \widehat{g}(x_F), & p_T &=& Z_k p_Z, \\ p_k &=& p_N + p_T, & U_k^T \widehat{\pi}_k &=& Y_k^T \widehat{g}(x_k + p_k), & & & \end{array} \tag{3.11}$$

where U_k is nonsingular and the columns of Z_k lie in the null space of J_k. These equations indicate that the QP step is the sum of the vectors p_N and p_T, where p_N is the Newton step to the linearized constraints and p_T is a quasi-Newton step based on *approximate* second-derivative information associated with the reduced Hessian $Z_k^T B_k Z_k$. Because of this disparity in the quality of the Newton steps, *the constraints tend to converge to zero faster than the reduced gradient* and the convergence of a quasi-Newton SQP method is characterized by the relationship $\|p_N\|/\|p_T\| \to 0$, i.e., the final search directions lie almost wholly in the null space of $J(x^*)$.

If x_k is far from a solution, the approximate curvature $y_k^T d_k$ may not be positive and the formula (3.8) will give an indefinite or undefined B_{k+1}. If, as in the case of unconstrained minimization, the update is skipped when $y_k^T d_k \le 0$, no new information about curvature of the Lagrangian will be gained. In this situation, an alternative pair of vectors satisfying $y_k^T d_k > 0$ can be used. Given the definition (3.9) of the least permissible approximate curvature, Powell [154] redefines y_k as $y_k + \varDelta y_k$, where $\varDelta y_k$ chosen so that $(y_k + \varDelta y_k)^T d_k = \sigma_k$, i.e.,

$$\varDelta y_k = \frac{\sigma_k - y_k^T d_k}{d_k^T (y_k - B_k d_k)} \big(y_k - B_k d_k \big).$$

The Powell modification is always well defined, which implies that it is always applied—even when it might be unwarranted because of negative curvature of the Lagrangian in the null space of $\widetilde{J}_k$ (cf. (3.7)).

3.2.2. Properties of the merit function

The Han-Powell merit function $\mathcal{M}(x;\rho) = P_1(x;\rho)$ has the appealing property that x^* is an unconstrained minimizer of $P_1(x;\rho)$ for $\rho > \|\pi^*\|_\infty$ (see, e.g., Zangwill [184], and Han and

Mangasarian [122]). A potential line-search model for $P_1(x;\rho)$ is

$$m_k(x;\rho) = f_k + g_k^T(x-x_k) + \tfrac{1}{2}(x-x_k)^T B_k(x-x_k) + \rho\|c_k + J_k(x-x_k)\|_1,$$

which is the ℓ_1 penalty function defined with local affine and quadratic approximations for c and f. However, because B_k is positive definite, a stronger condition on α is defined by omitting the quadratic term and using the line-search model

$$m_k(x;\rho) = f_k + g_k^T(x-x_k) + \rho\|c_k + J_k(x-x_k)\|_1. \tag{3.12}$$

To obtain a smaller value of $P_1(x;\rho)$ at each iteration, the line-search model must satisfy $\mathit{\Delta} m_k(d_k;\rho) > 0$, where $\mathit{\Delta} m_k(d;\rho)$ is the predicted reduction in $\mathcal{M}$ analogous to (3.2). The optimality conditions (2.18) for the QP subproblem together with the affine model (3.12) defined with $x = x_k + \alpha p_k$ allow us to write the predicted reduction as

$$\begin{aligned}\mathit{\Delta} m_k(\alpha p_k;\rho) &= \alpha\big(\rho\|c_k\|_1 + c_k^T\widehat{\pi}_k - p_k^T\widehat{z}_k + p_k^T B_k p_k\big)\\ &= \alpha\big(\rho\|c_k\|_1 + c_k^T\widehat{\pi}_k - (\widehat{x}_k - x_k)^T\widehat{z}_k + p_k^T B_k p_k\big).\end{aligned} \tag{3.13}$$

The QP optimality conditions give $\widehat{x}_k \cdot \widehat{z}_k = 0$, yielding

$$\begin{aligned}\mathit{\Delta} m_k(\alpha p_k;\rho) &= \alpha\big(\rho\|c_k\|_1 + c_k^T\widehat{\pi}_k + x_k^T\widehat{z}_k + p_k^T B_k p_k\big)\\ &\ge \alpha\Big(\sum_{i=1}^m |c_i(x_k)|\big(\rho - |(\widehat{\pi}_k)_i|\big) + \|x_k \cdot \widehat{z}_k\|_1 + p_k^T B_k p_k\Big).\end{aligned}$$

This inequality implies that if B_k is positive definite, then a sufficient condition for the inequality $\mathit{\Delta} m_k(\alpha p_k;\rho) > 0$ to hold is $\rho \ge \|\widehat{\pi}_k\|_\infty$. Han [120] uses this condition to define a nondecreasing sequence $\{\rho_k\}$ such that $\rho_k > \|\widehat{\pi}_j\|_\infty$ for all $k \ge j$. With this definition of $\{\rho_k\}$, and under assumptions that include the uniform boundedness of the sequence $\{B_k\}$ and the existence of at least one nonnegative x such that $c_k + J_k(x-x_k) = 0$, Han shows that all accumulation points of the sequence $\{x_k\}$ are first-order KKT points of the constrained problem (1.2).

3.2.3. Extensions

The introduction of the Wilson-Han-Powell SQP method (i.e., the plain SQP method with a convex subproblem and a line-search with respect to a merit function) had an immediate beneficial effect on the performance of optimization codes. However, as is the case with all successful innovations, it was not long before certain issues were identified that have an impact on performance and reliability. In this section we consider some of these issues and outline some extensions of the Wilson-Han-Powell method that are intended to address them.

The Maratos effect and alternative merit functions. The value of the penalty parameter in the ℓ_1 merit function $\mathcal{M}$ in (3.5) can have a substantial effect on the overall efficiency of an SQP method. When solving a sequence of related problems, it may be possible to provide a good estimate of the optimal multipliers, and a value of $\rho \approx \|\pi^*\|_\infty$ can be specified. When ρ is large relative to the magnitude of f, the level surfaces of $\mathcal{M}$ closely resemble the constraint surface $c(x) = 0$. If the constraints are changing rapidly and the SQP outer iterates become close to a nonoptimal point near the constraints (as is the case for methods that use a quasi-Newton approximation B_k, see Section 3.2.1), the iterates must negotiate the base of a steep-sided curved valley. In this situation, the affine model of the constraints provides for only a limited amount of progress along the SQP direction, and the step $\alpha = 1$ fails to reduce the value of $\mathcal{M}$. This rejection of the plain SQP step near x^* causes a breakdown of the superlinear convergence rate. Various strategies have been devised to prevent this phenomenon, which is known as the "Maratos effect" (see Maratos [136]). One approach is to use a "nonmonotone" line search that allows the merit function to increase for a limited number of iterations (see, e.g., Chamberlain et al. [31], and Dai and Schittkowski [46]).

Another approach, proposed by Fletcher [63], seeks to reduce the magnitude of the penalty term by computing a *second-order correction* to the Newton step. The second-order correction uses the step $p_k + s_k$, where the step s_k is the solution of a second subproblem:

$$\begin{array}{ll} \underset{s\in\mathbb{R}^n}{\text{minimize}} & f_k + g_k^T(p_k+s) + \tfrac{1}{2}(p_k+s)^T B_k(p_k+s) \\ \text{subject to} & c(x_k+p_k) + J_k s = 0, \quad x_k + p_k + s \ge 0. \end{array} \tag{3.14}$$

The second-order correction requires an additional constraint evaluation at the SQP point $x_k + p_k$.

If a feasible-point active-set method is used to solve (3.6) and the subproblem is started with the basic set from the previous iteration, the second-order correction need be computed only if (3.6) is solved in one iteration, which is a necessary condition for the method to be in the final stages of convergence. If the solution of (3.14) is also identified in one iteration and $\widetilde{J}_k$ is the matrix of basic columns of J_k, then s_k satisfies the equations

$$\begin{pmatrix} \widetilde{B}_k & \widetilde{J}_k^T \\ \widetilde{J}_k & 0 \end{pmatrix} \begin{pmatrix} s_k \\ -\bar{\pi}_k \end{pmatrix} = -\begin{pmatrix} \big(g_k + B_k(p_k+\eta_k)\big)_{\mathcal{B}} \\ c(x_k+p_k) + J_k\eta_k \end{pmatrix}, \tag{3.15}$$

where $\bar{\pi}_k$ is the vector of optimal multipliers for (3.14) and η_k is defined as in (2.21). In this situation, if some factorization of the KKT matrix is available on termination of the solution of (3.6), the correction may be obtained with just one solve with a different right-hand side. For an analysis of the rate of convergence, see Fletcher [63] and Yuan [182].

Other merit functions may be defined for different choices of the norm of the constraint violations. For the infinity-norm, the ℓ_∞ penalty function $P_\infty(x;\rho)$ defined in (1.9) may be used in conjunction with the line-search model

$$m_k(x;\rho) = f_k + g_k^T(x-x_k) + \rho\|c_k + J_k(x-x_k)\|_\infty.$$

This model predicts a reduction in $P_\infty(x;\rho)$ if $p_k^T B_k p_k \ge 0$ and $\rho > \|\widehat{\pi}_k\|_1$. Anitescu [3] considers the convergence of the ℓ_∞ merit function when applied with various line-search strategies and a convex QP subproblem.

Like its ℓ_1 counterpart, the ℓ_∞ penalty function can exhibit the Maratos effect for large values of ρ. Merit functions that do not have this problem may be defined by using a smooth norm for the constraint violations. In general, a merit function may be defined in terms of the primal variables only, or may include estimates of the Lagrange multipliers. A merit function that does not suffer from the Maratos effect is the augmented Lagrangian function:

$$\mathcal{M}(x,\pi;\rho) \equiv f(x) - \pi^T c(x) + \tfrac{1}{2}\rho c(x)^T c(x), \tag{3.16}$$

where π is a multiplier estimate and ρ is a nonnegative penalty parameter. Schittkowski [165, 166, 167], and Gill et al. [95, 85] define SQP methods in which both the primal and dual variables are modified by the line search, with

$$x_{k+1} = x_k + \alpha_k p_k, \qquad \pi_{k+1} = \pi_k + \alpha_k q_k, \tag{3.17}$$

where the primal-dual search directions $p_k = \widehat{x}_k - x_k$ and $q_k = \widehat{\pi}_k - \pi_k$ are based on the solution $(\widehat{x}_k, \widehat{\pi}_k)$ of a convex QP with a quasi-Newton Hessian. When an augmented

Lagrangian is used in the conventional role as an objective function for sequential unconstrained minimization, new multiplier estimates are obtained by *maximizing* with respect to the dual variables. In the SQP context, the inclusion of the dual variables as arguments for *minimization* serves to make the augmented Lagrangian a continuous function of both the primal and dual variables, with the step length acting as a continuation parameter that links the old and new values of π. If necessary, the penalty parameter is increased to ensure that the primal-dual direction is a descent direction for the merit function. However, it can be shown that under typical assumptions on the problem, the penalty parameter remains bounded (see Gill et al. [95], and Murray and Prieto [144] for details). If the objective is convex and the feasible region is a convex set, it is often the case that the penalty parameter never needs to be increased from an initial value of zero.

A number of line-search SQP methods have been proposed that use variants of the conventional augmented Lagrangian as a merit function (see, e.g., DiPillo and Grippo [50], Bertsekas [7], Byrd, Tapia and Zhang [30], and Anitescu [3]). A primal-dual augmented Lagrangian has been proposed by Gill and Robinson [97]. Given an estimate π_E of the multipliers π^*, consider the function

$$\begin{aligned} L_A(x,\pi;\pi_E,\mu) = f(x) - c(x)^T\pi_E + \frac{1}{2\mu}\|c(x)\|_2^2 \\ + \frac{1}{2\mu}\|c(x)+\mu(\pi-\pi_E)\|^2, \end{aligned} \tag{3.18}$$

where μ is a positive inverse penalty parameter (see also, Forsgren and Gill [72], Robinson [157], and Gill and Robinson [97]). The primal-dual augmented Lagrangian has a bound-constrained minimization property analogous to the conventional augmented Lagrangian (1.11). In particular, if π_E is given the value of the optimal multiplier vector π^*, then (x^*,π^*) is a first-order KKT point for the bound-constrained problem

$$\underset{x\in\mathbb{R}^n;\ \pi\in\mathbb{R}^m}{\text{minimize}} \quad L_A(x,\pi;\pi^*,\mu) \quad \text{subject to} \quad x \ge 0.$$

Moreover, if the second-order sufficient conditions for optimality hold, then there exists a finite $\bar{\mu}$ such that (x^*,π^*) is an isolated unconstrained minimizer of L_A for all $\mu < \bar{\mu}$. It follows that L_A may be minimized simultaneously with respect to both the primal and dual variables. A benefit of using L_A as an SQP merit function is that it may be used in conjunction with a regularized method for solving the QP subproblem (see Section A.3 of the Appendix for details).

We conclude this section by mentioning methods that avoid the need for a merit function altogether by generating iterates that are always feasible. In many physical and engineering applications, the constraint functions not only characterize the desired properties of the solution, but also define a region in which the problem statement is meaningful (for example, $f(x)$ or some of the constraint functions may be undefined outside the feasible region). In these applications, an interior point can usually be determined trivially. Interior methods are therefore highly appropriate for this class of problem. However, several SQP methods have been proposed for optimization in this context, see, e.g., Lawrence and Tits [129], and Kostreva and Chen [126,127]. These methods are suitable for problems that have only inequality constraints, the only exception being *linear* equality constraints, which can be kept feasible at every iterate (see, e.g., Gill, Murray and Wright [96]).

Formulation of the QP subproblem. A potential difficulty associated with SQP methods based on the direct linearization of the nonlinear constraints, is that the QP subproblem may be infeasible. This can be caused by the nonlinear constraints being infeasible, or by a poor linearization at the current iterate. In the context of the Wilson-Han-Powell method, this problem may be resolved by perturbing the QP subproblem so that the constraints always have a feasible point. The magnitude of the perturbation is then reduced or eliminated as the iterates converge. Powell [155] focused on the case of an infeasible linearization and considered the modified QP:

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n,\theta\in\mathbb{R}^1}{\text{minimize}} & f_k + g_k^T(x-x_k) + \tfrac{1}{2}(x-x_k)^T B_k(x-x_k) + \tfrac{1}{2}\rho_k(1-\theta)^2 \\ \text{subject to} & (1-\theta)c_k + J_k(x-x_k) = 0, \quad x + \theta\,[\,x_k\,]_- \ge 0, \end{array} \tag{3.19}$$

where $[\,x\,]_-$ is the vector with components $\max\{-x_i, 0\}$, and θ is an additional variable that is driven to zero by increasing the nonnegative penalty parameter ρ_k. The modified QP is always feasible—for example, the point $(x,\theta) = (x_k, 1)$ satisfies the constraints.

Burke [22, 23], and Burke and Han [24] use a different approach that is based on the observation that the problem (1.2) is actually two problems in one: the *feasibility problem* of satisfying the constraints, and the *optimality problem* of minimizing f. They define a line-search algorithm that has the primary goal of attaining feasibility.

The computation of the line-search direction is organized into two phases. The first phase ignores the objective and computes a descent direction for a function that measures the distance of an arbitrary point to the set of feasible points for the nonlinear problem. The required direction is computed by minimizing the distance to the feasible set for the *linearized* constraints. For the second phase, the constraint residuals corresponding to the optimal value of the distance function are used to modify the constraints of the conventional QP subproblem. The modified QP is always feasible, and the resulting direction is used in a line search with a merit function that includes a term involving the value of the distance function. Under certain assumptions, this procedure provides a sequence that converges to a first-order stationary point of either the original problem or the distance function.

The definition of the distance function requires a choice of norm, although Burke and Han provide a general analysis that is independent of the norm. For simplicity, we describe the computations for each phase when the distance function is defined in terms of the one-norm. Given current values of parameters σ_k and β_k such that $0 < \sigma_k \le \beta_k$, the first phase involves the solution of the linear program

$$\begin{array}{ll} \underset{x,v\in\mathbb{R}^n;\; u\in\mathbb{R}^m}{\text{minimize}} & e^T u + e^T v \\ \text{subject to} & -u \le c_k + J_k(x-x_k) \le u, \quad x + v \ge 0, \quad v \ge 0, \\ & -\sigma_k e \le x - x_k \le \sigma_k e. \end{array} \tag{3.20}$$

This problem gives vectors u and v of least one-norm for which the constraints $c_k + J_k(x - x_k) = u$, $x + v \ge 0$ and $\|x - x_k\|_\infty \le \sigma_k$ are feasible. If the original linearized constraints are feasible, then the work necessary to solve problem (3.20) is comparable to that of the feasibility phase of a two-phase active-set method for the plain QP subproblem (see Section A.1). The difference is the extra expense of locating a bounded feasible point with *least-length* distance from x_k. Let x_F denote the computed solution of the phase-1 problem (3.20). The computation for phase 2 involves the solution of the QP:

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f_k + g_k^T(x-x_k) + \tfrac{1}{2}(x-x_k)^T B_k(x-x_k) \\ \text{subject to} & c_k + J_k(x-x_k) = \widehat{c}_k(x_F), \quad x + [\,x_F\,]_- \ge 0, \\ & -\beta_k e \le x - x_k \le \beta_k e, \end{array} \tag{3.21}$$

where, as usual, $\widehat{c}_k(x)$ denotes the vector of linearized constraint functions $\widehat{c}_k(x) = c_k + J_k(x - x_k)$. The phase-2 problem (3.21) is a convex program with bounded solution $\widehat{x}_k$ (say). This solution is used to define the search direction $p_k = \widehat{x}_k - x_k$ for a line search on the merit function

$$\mathcal{M}(x) = f(x) + \rho_k\|c(x)\|_1 + \rho_k\|\,[\,x\,]_-\,\|_1.$$

Once $x_{k+1} = x_k + \alpha_k p_k$ has been determined, the positive-definite approximate Hessian B_k is updated as in the conventional Han-Powell method. For details on how the parameters ρ_k, σ_k and β_k are updated, the reader is referred to Burke and Han [24]. Other related variants of the Wilson-Han-Powell formulation are proposed by Liu and Yuan [132], and Mo, Zhang and Wei [137].

Another approach to the treatment of infeasibility is to modify the original *nonlinear* constraints so that the linearized constraints of the QP subproblem are always feasible. This is the approach taken in the method of SNOPT (see Gill, Murray and Saunders [85]). The method proceeds to solve (1.2) as given, using QP subproblems based on the conventional linearization of the nonlinear constraints. If a QP subproblem proves to be infeasible or unbounded (or if the Lagrange multiplier estimates become large), SNOPT enters *nonlinear elastic mode* and switches to the nonlinear elastic problem (1.4). The QP subproblem for the nonlinear elastic problem is given by

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n;\; u,v\in\mathbb{R}^m}{\text{minimize}} & \widehat{f}_k(x) + \rho_k e^T u + \rho_k e^T v \\ \text{subject to} & c_k + J_k(x - x_k) - u + v = 0, \\ & x \ge 0, \quad u \ge 0, \quad v \ge 0, \end{array} \tag{3.22}$$

where $\widehat{f}_k(x) + \rho_k e^T u + \rho_k e^T v$ is the *composite QP objective* and ρ_k is a nonnegative *penalty parameter* or *elastic weight* analogous to the quantity defined in Section 1.3. This problem is always feasible and the solution is used in conjunction with a merit function defined in terms of the nonlinear elastic problem (1.4).

Quasi-Newton updates and indefiniteness. A necessary condition for Q-superlinear convergence is that the approximate Hessian matrices $\{B_k\}$ satisfy

$$\lim_{k\to\infty} \frac{\|Z_k Z_k^T\big(B_k - H(x^*,\pi^*)\big)Z_k Z_k^T d_k\|}{\|d_k\|} = 0,$$

where Z_k is the matrix defined in (3.11) (see Boggs and Tolle [10]). The definition of y_k and d_k should ensure that this condition is satisfied as the solution is approached, so that Q-superlinear convergence is not inhibited.

One possible modification uses the intermediate point x_F defined by equations (3.11). If x_F is known, new values of d_k and y_k are computed based on evaluating the nonlinear functions at the point $w_k = x_k + \alpha_k(x_F - x_k)$. The BFGS update is then attempted using the update pair:

$$d_k = x_{k+1} - w_k, \qquad y_k = \nabla_x L(x_{k+1}, \pi_{k+1}, z_{k+1}) - \nabla_x L(w_k, \pi_{k+1}, z_{k+1}).$$

The purpose of this modification is to exploit the properties of the reduced Hessian in the neighborhood of a local minimizer of (2.4). With this choice of w_k, the change in variables is $d_k = x_{k+1} - w_k = \alpha_k p_T$, where p_T is the vector $\widehat{x}_k - x_F$ (see (3.11) above). Then,

$$y_k^T d_k = \alpha_k y_k^T p_T \approx \alpha_k^2 p_T^T H(x_k, \pi_{k+1}) p_T = \alpha_k^2 p_Z^T Z_k^T H(w_k, \pi_{k+1}) Z_k p_Z.$$

It follows that $y_k^T d_k$ approximates the curvature of the reduced Hessian, which is positive definite sufficiently close to an isolated local minimizer of (2.4). If this modification does not provide sufficiently positive approximate curvature, no update is made. An additional function evaluation is required at w_k, but the modification is rarely needed more than a few times—even when the Hessian of the Lagrangian has negative eigenvalues at a solution. (For further information, see Gill, Murray and Saunders [85].)

Large-scale Hessians. If the number of variables is large, conventional quasi-Newton methods are prohibitively expensive because of the need to store the (dense) matrix B_k. A limited-memory approach uses a fixed number of vectors, say ℓ, to define a positive-definite approximation to $H(x_k, \pi_k)$ based on curvature information accrued during the most recent ℓ iterations. Let ℓ be preassigned (say $\ell = 10$), and consider any iteration k such that $k \ge \ell - 1$. Given any initial positive-definite approximation $B_k^{(0)}$ to $H(x_k, \pi_k)$, consider the sequence of matrices $\{B_k^{(i)}\}$, for $i = k - \ell,\ k - \ell + 1,\ \dots,\ k$, such that

$$B_k^{(k-\ell)} = B_k^{(0)}, \quad B_k^{(i+1)} = B_k^{(i)} + v_i v_i^T - u_i u_i^T, \quad i = k - \ell, \dots, k - 1,$$

where the $\{(u_i, v_i)\}$ are ℓ vector pairs with each pair (u_i, v_i) defined in terms of $(d_{k-\ell}, y_{k-\ell})$, $\dots$, (d_{i-1}, y_{i-1}) (cf. (3.10)) via

$$u_i = \frac{1}{(d_i^T B_k^{(i)} d_i)^{\frac{1}{2}}} B_k^{(i)} d_i, \quad \text{and} \quad v_i = \frac{1}{(y_i^T d_i)^{\frac{1}{2}}}\, y_i.$$

Similar limited-memory quasi-Newton methods are described by Nocedal and Wright [145], Buckley and LeNir [16, 17] and Gilbert and Lemaréchal [78]. More elaborate schemes are given by Liu and Nocedal [131], Byrd, Nocedal, and Schnabel [28], and Gill and Leonard [81], and some have been evaluated by Morales [138].

The definition of B_k requires the ℓ pairs (u_i, v_i). Each of the vectors u_i $(k-\ell \le i \le k-1)$ involves the product $B_k^{(i)} d_i$, which is computed using the recurrence relation

$$B_k^{(i)} d_i = B_k^{(0)} d_i + \sum_{j=k-\ell}^{i-1} \big((v_j^T d_i) v_j - (u_j^T d_i) u_j \big).$$

For the vectors v_i $(k - \ell \le i \le k - 1)$ and scalars $v_j^T d_i$ $(k - \ell \le j \le i - 1)$, only v_{k-1} and $v_j^T d_{k-1}$ $(k - \ell \le j \le k - 2)$ need to be computed at iteration k as the other quantities are available from the previous iteration.

A separate calculation may be used to update the *diagonals* of B_k from (3.8). On completion of iteration k, these diagonals form the next positive-definite $B_{k+1}^{(0)}$. Then, at the kth iteration, we define the approximate Hessian

$$B_k = B_k^{(k)} = B_k^{(0)} + V_k V_k^T - U_k U_k^T,$$

where $U_k = \begin{pmatrix} u_{k-\ell} & u_{k-\ell+1} & \cdots & u_{k-1} \end{pmatrix}$ and $V_k = \begin{pmatrix} v_{k-\ell} & v_{k-\ell+1} & \cdots & v_{k-1} \end{pmatrix}$. It must be emphasized that B_k is not computed explicitly. Many sparse QP solvers access B_k by requesting products of the form $B_k u$. These are computed with work proportional to ℓ. For situations where the QP solver solves an explicit sparse system of the form (3.7), the solution may be found using the bordered matrix

$$\begin{pmatrix} \widetilde{B}_k^{(0)} & \widetilde{J}_k^T & \widetilde{V}_k^T & \widetilde{U}_k^T \\ \widetilde{J}_k & & & \\ \widetilde{V}_k & & I & \\ \widetilde{U}_k & & & -I \end{pmatrix} \begin{pmatrix} p_k \\ -\widehat{\pi}_k \\ r \\ s \end{pmatrix} = - \begin{pmatrix} (g_k + B_k \eta_k)_{\mathcal{B}} \\ c_k + J_k \eta_k \\ 0 \\ 0 \end{pmatrix},$$

where $\widetilde{B}_k^{(0)}$, $\widetilde{J}_k$, $\widetilde{V}_k$ and $\widetilde{U}_k$ denote the matrices of basic components of $B_k^{(0)}$, J_k, V_k and U_k. Following [90, Section 3.6.2], if we define

$$K_0 = \begin{pmatrix} \widetilde{B}_k^{(0)} & \widetilde{J}_k^T \\ \widetilde{J}_k & \end{pmatrix}, \qquad S = \begin{pmatrix} I & \\ & -I \end{pmatrix} - \begin{pmatrix} \widetilde{V}_k^T \\ \widetilde{U}_k^T \end{pmatrix} K_0^{-1} \begin{pmatrix} \widetilde{V}_k & \widetilde{U}_k \end{pmatrix},$$

it would be efficient to work with a sparse factorization of K_0 and dense factors of its Schur complement S. (For a given QP subproblem, U and V are constant, but changes to $\widetilde{J}_k$ would be handled by appropriate updates to the Schur complement. See Section A.4.2 of the Appendix.) For general QP solvers that require an explicit sparse Hessian, the limited-memory updates can be applied implicitly by including additional linear equality constraints in the QP subproblem, see Gould and Robinson [109]. Bradley [13] describes a BFGS limited-memory method for SQP that employs a diagonal approximation in conjunction with a circular buffer.

In practice, the quasi-Newton approximation may become indefinite because of rounding error and it is better numerically to write B_k in the form $B_k = G_k^T G_k$, where G_k is the product of elementary matrices

$$G_k = G_k^{(0)} \prod_{j=k-\ell}^{k-1} (I + d_j w_j^T), \tag{3.23}$$

with $B_k^{(0)} = G_k^{(0)T} G_k^{(0)}$ and $w_j = (\pm v_j - u_j)/(d_j^T B_k^{(j)} d_j)^{\frac{1}{2}}$ (see Brodlie, Gourlay and Greenstadt [14], Dennis and Schnabel [49], and Gill, Murray and Saunders [85]). The sign of v_j may be chosen to minimize the rounding error in computing w_j. The quantities (d_j, w_j) are stored for each j. During outer iteration k, the QP solver accesses B_k by requesting products of the form $B_k z$. These are computed with work proportional to ℓ using the recurrence relations:

$$\begin{array}{ll} z \leftarrow z + (w_j^T z) d_j,\ j = k-1 : k-\ell; & z \leftarrow G_k^{(0)} z; \\ t \leftarrow G_k^{(0)T} z; & t \leftarrow t + (d_j^T t)\, w_j,\ j = k-\ell : k-1. \end{array}$$

Products of the form $u^T B_k u$ are easily and safely computed as $\|z\|_2^2$ with $z = G_k u$.

In a QP solver that updates the Schur complement matrix an explicit sparse Hessian, the system (3.7) with $B_k = G_k^T G_k$ is equivalent to

$$\left(\begin{array}{cc|ccccc} \widetilde{B}_k^{(0)} & \widetilde{J}_k^T & \widetilde{u}_{k-\ell} & \widetilde{w}_{k-\ell} & \cdots & \widetilde{u}_{k-1} & \widetilde{w}_{k-1} \\ \widetilde{J}_k & & & & & & \\ \hline \widetilde{u}_{k-\ell}^T & & \gamma_{k-\ell} & -1 & & & \\ \widetilde{w}_{k-\ell}^T & & -1 & & & & \\ \vdots & & & & \ddots & & \\ \widetilde{u}_{k-1}^T & & & & & \gamma_{k-1} & -1 \\ \widetilde{w}_{k-1}^T & & & & & -1 & \end{array}\right) \begin{pmatrix} p_k \\ -\widehat{\pi}_k \\ r_{k-\ell} \\ s_{k-\ell} \\ \vdots \\ r_{k-1} \\ s_{k-1} \end{pmatrix} = - \begin{pmatrix} (g_k + B_k \eta_k)_{\mathcal{B}} \\ c_k + J_k \eta_k \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix},$$

where $\widetilde{d}_j = (d_j)_{\mathcal{B}}$, $\widetilde{u}_j = (B_k^{(j)} d_j)_{\mathcal{B}}$, and $\gamma_j = \widetilde{d}_j^T \widetilde{u}_j$ (see Gill, Murray and Saunders [85], and Huynh [125]).

An alternative form of the limited-memory update is used by Gill, Murray and Saunders [85]. Let r and k denote two outer iterations such that $r \le k \le r+\ell$. At iteration k the BFGS approximate Hessian may be expressed in terms of ℓ updates to a positive-definite B_r:

$$B_k = B_r + \sum_{i=r}^{k-1} \left(v_i v_i^T - u_i u_i^T\right), \tag{3.24}$$

where $u_i = B_i d_i / (d_i^T B_i d_i)^{\frac{1}{2}}$, and $v_i = y_i / (y_i^T d_i)^{\frac{1}{2}}$. In this scheme, the $k - r$ pairs (u_i, v_i) do not need to be recomputed for each update. On completion of iteration $k = r + \ell$, a total of ℓ

pairs have been accumulated, and the storage is "reset" by discarding the previous updates. Moreover, the definition of u_i is simplified by the identity $B_i d_i = -\alpha_i \nabla_x L(\widehat{x}_i, \widehat{\pi}_i, \widehat{z}_i)$ that follows from the QP optimality conditions (2.18). As in the previous scheme, a separate calculation may be used to update the diagonals of B_k from (3.8). On completion of iteration $k = r + \ell$, these diagonals form the next positive-definite B_r (with $r = k + 1$).

This scheme has an advantage in the SQP context when the constraints are linear: the reduced Hessian for the QP subproblem can be updated between outer iterations (see Section A.4.1).

Early termination of QP subproblems. SQP theory usually assumes that the QP subproblems are solved to optimality. For large problems with a poor starting point and $B_0 = I$, many thousands of iterations may be needed for the first QP, building up many free variables that are promptly eliminated by more thousands of iterations in the second QP. In general, it seems wasteful to expend much effort on any QP before updating B_k and the constraint linearization.

Any scheme for early termination must be implemented in a way that does not compromise the reliability of the SQP method. For example, suppose that the QP iterations are terminated after an arbitrary fixed number of steps. If a primal active-set method is used to solve the subproblem, the multipliers associated with QP constraints that have not been optimized will be negative. Using these multipliers directly (or first setting them to zero) in the definition of the Lagrangian function is problematic. The resulting search direction may not be a descent direction for the merit function, or may require the penalty parameter to be increased unnecessarily. For example, the value of the lower bound on the penalty parameter for the ℓ_1 merit function involves the values of the QP multipliers—see, (3.13). Dembo and Tulowitzki [48] suggest using a dual feasible active-set method for the QP subproblem and terminating the inner iterations when the norm of a potential search direction $p_k = \widehat{x}_k - x_k$ is small. Dual feasible active-set methods have the advantage that the approximate multipliers are nonnegative, but a terminated iteration will have some negative primal variables—this time making the definition of the search direction problematic.

Murray and Prieto [144] suggest another approach to terminating the QP solutions early, requiring that at least one QP subspace stationary point be reached (see Definition A.1 of the Appendix). The associated theory implies that any subsequent point $\widehat{x}_k$ generated by a special-purpose primal-feasible QP solver gives a sufficient decrease in the augmented Lagrangian merit function (3.16), provided that $\|\widehat{x}_k - x_k\|$ is nonzero.

Another way to save inner iterations safely during the early outer iterations is to *suboptimize* the QP subproblem. At the start of an outer iteration, many variables are fixed at their current values (i.e., x_i is fixed at $(x_k)_i$) and an SQP outer iteration is performed on the reduced problem (solving a smaller QP to get a search direction for the nonfixed variables). Once a solution of the reduced QP is found, the fixed variables are freed, and the outer iteration is completed with a "full" search direction that happens to leave many variables unaltered because $p_i = (\widehat{x}_i - x_k)_i = 0$ for the temporarily fixed variables. At each step, the conventional theory for the reduction in the merit function should guarantee progress on the associated reduced *nonlinear* problem. In practice, it may not be obvious which variables should be fixed at each stage, the reduced QP could be infeasible, and degeneracy could produce a zero search direction. Instead, the choice of which variables to fix is made within the QP solver. In the method of SNOPT, QP iterations are performed on the full problem until a feasible point is found or elastic mode is entered. The iterations are continued until certain limits are reached and not all steps have been degenerate. At this point all variables such that $x_i = (x_k)_i$ are frozen at their current value and the reduced QP is solved to optimality. With this scheme it is safe to impose rather arbitrary limits, such as limits on the number of iterations (for the various termination conditions that may be applied, see Gill, Murray and Saunders [85, 86]). Note that this form of suboptimization enforces the condition $((\widehat{x}_k - x_k) \cdot \widehat{z}_k)_i = 0$ for the frozen variables and so the nonoptimized variables have no affect on the magnitude of the penalty parameter in (3.13).

3.3. Sequential unconstrained methods

Fletcher [62] observed that the ℓ_1 penalty function (1.8) can be minimized subject to bounds by solving a sequence of nondifferentiable *unconstrained* subproblems of the form

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \widehat{f}_k(x) + \rho\|\widehat{c}_k(x)\|_1 + \rho\|\,[\,x\,]_-\,\|_1, \tag{3.25}$$

where $\widehat{c}_k(x)$ denotes the linearized constraint functions $\widehat{c}_k(x) = c_k + J_k(x - x_k)$, and $[\,v\,]_- = \max\{-v_i, 0\}$. In this case the bound constraints are *not* imposed explicitly. Fletcher proposed minimizing this function using a trust-region method, although a line-search method would also be appropriate, particularly if H_k were positive definite. The trust-region subproblem has the form

$$\begin{array}{ll} \underset{d\in\mathbb{R}^n}{\text{minimize}} & \widehat{f}_k(x_k+d) + \rho\|\widehat{c}_k(x_k+d)\|_1 + \rho\|\,[\,x_k+d\,]_-\,\|_1 \\ \text{subject to} & \|d\| \le \delta_k. \end{array} \tag{3.26}$$

The trust-region radius δ_k is increased or decreased based on condition (3.5), where $\mathcal{M}(x;\rho)$ is the penalty function $P_1(x;\rho)$ and $m_k(x;\rho)$ is defined in terms of an affine or quadratic model of the modified Lagrangian (see (3.12)). This approach forms the basis of Fletcher's "Sℓ_1-QP method". Each subproblem has a piecewise quadratic objective function and has the same complexity as a quadratic program of the form (2.1). If the infinity norm is used to define the size of the trust region, the subproblem is equivalent to the smooth quadratic program

$$\begin{array}{ll} \underset{d,w\in\mathbb{R}^n;\; u,v\in\mathbb{R}^m}{\text{minimize}} & \widehat{f}_k(x_k+d) + \rho e^Tu + \rho e^Tv + \rho e^Tw \\ \text{subject to} & \widehat{c}_k(x_k+d) - u + v = 0, \quad u \ge 0, \quad v \ge 0, \\ & -\delta_k e \le d \le \delta_k e, \quad x_k + d + w \ge 0, \quad w \ge 0, \end{array}$$

where, analogous to (1.4), the vectors u and v may be interpreted as the positive and negative parts of the affine function $c_k + J_k d$. A benefit of this formulation is that a solution of (3.26) always exists, even when the linearized constraints of the plain QP subproblem are inconsistent. Observe that the unconstrained subproblem (3.25) is defined in terms of $\widehat{f}_k(x)$, the model of the modified Lagrangian (see (2.2)). This feature is crucial because it implies that if the trust-region radius and penalty parameter are sufficiently large in the neighborhood of an isolated solution, the Sℓ_1-QP subproblem is the same as the plain SQP subproblem (2.1). Nevertheless, the implicit minimization of the ℓ_1 penalty function means that there is the possibility of the Maratos effect. For the trust-region approach, the second-order correction may be determined from the quadratic program

$$\begin{array}{ll} \underset{s,w\in\mathbb{R}^n;\; u,v\in\mathbb{R}^m}{\text{minimize}} & \widehat{f}_k(x_k+d_k+s) + \rho e^Tu + \rho e^Tv + \rho e^Tw \\ \text{subject to} & J_k s - u + v = -c(x_k+d_k), \quad u \ge 0, \quad v \ge 0, \\ & -\delta_k e \le d_k + s \le \delta_k e, \quad x_k + d_k + s \ge 0, \quad w \ge 0. \end{array} \tag{3.27}$$

Yuan [182] gives an analysis of the superlinear convergence of trust-region methods that use the second-order correction.

The Sℓ_1-QP approach can be used in conjunction with other unconstrained merit functions. Many of these extensions lead to a subproblem that is equivalent to a quadratic program. The "Sℓ_∞-QP method" uses the trust-region subproblem

$$\begin{array}{ll} \underset{d\in\mathbb{R}^n}{\text{minimize}} & \widehat{f}_k(x_k+d) + \rho\|\widehat{c}_k(x_k+d)\|_\infty + \rho\|\,[\,x_k+d\,]_-\,\|_\infty \\ \text{subject to} & \|d\|_\infty \le \delta_k, \end{array} \tag{3.28}$$

which is equivalent to the quadratic program

$$\begin{array}{ll} \underset{d\in\mathbb{R}^n;\ \theta,\sigma\in\mathbb{R}}{\text{minimize}} & \widehat{f}_k(x_k+d)+\rho\theta+\rho\sigma \\ \text{subject to} & -\theta e \le \widehat{c}_k(x_k+d) \le \theta e, \quad \theta \ge 0, \\ & -\delta_k e \le d \le \delta_k e, \quad x_k+d+\sigma e \ge 0, \quad \sigma \ge 0, \end{array} \tag{3.29}$$

see, e.g., Yuan [183], and Exler and Schittkowski [58]. A QP problem for the second-order correction may be defined analogous to (3.27). For a general discussion of the convergence properties of nondifferentiable exact penalty functions in the SQP context, see Fletcher [65], Burke [23], and Yuan [181].

3.4. Filter methods

The definition of the merit function in the Han-Powell method or the nonsmooth objective function in the sequential unconstrained optimization method requires the specification of a penalty parameter that weights the effect of the constraint violations against the value of the objective. Another way of forcing convergence is to use a *filter*, which is a two-dimensional measure of quality based on $f(x)$ and $\|c(x)\|$, where we assume that $x \ge 0$ is satisfied throughout. A filter method requires that progress be made with respect to the two-dimensional function $\big(\|c(x)\|, f(x)\big)$. Using the conventional notation of filter methods, we define $h(x) = \|c(x)\|$ as the measure of infeasibility of the equality constraints, and use (h_j, f_j) to denote the pair $\big(h(x_j), f(x_j)\big)$.

The two-dimensional measure provides the conditions for a point $\bar{x}$ to be "better" than a point $\widehat{x}$. Given two points $\bar{x}$ and $\widehat{x}$, the pair $\big(h(\bar{x}), f(\bar{x})\big)$ is said to *dominate* the pair $\big(h(\widehat{x}), f(\widehat{x})\big)$ if

$$h(\bar{x}) \le \beta h(\widehat{x}) \quad \text{and} \quad f(\bar{x}) \le f(\widehat{x}) - \gamma h(\bar{x}),$$

where β, $\gamma \in (0,1)$ are constants with $1-\beta$ and γ small (e.g., $\beta = 1-\gamma$ with $\gamma = 10^{-3}$). (For brevity, we say that $\bar{x}$ dominates $\widehat{x}$, although it must be emphasized that only the objective value and constraint norm are stored.) A filter $\mathcal{F}$ consists of a list of entries (h_j, f_j) such that no entry dominates another. (This filter is the so-called *sloping filter* proposed by Chin [32] and Chin and Fletcher [33]. The original filter proposed by Fletcher and Leyffer [67,68] uses $\gamma = 0$ and $\beta = 1$.)

A pair $\big(h(x_k), f(x_k)\big)$ is said to be "acceptable to the filter" $\mathcal{F}$ if and only if it is not dominated by any entry in the filter, i.e.,

$$h(x_k) \le \beta h_j \quad \text{or} \quad f(x_k) \le f_j - \gamma h(x_k) \tag{3.30}$$

for every $(h_j, f_j) \in \mathcal{F}$. In some situations, an accepted point $(h(x_k), f(x_k))$ is added to the filter. This operation adds $(h(x_k), f(x_k))$ to the list of entries (h_j, f_j) in $\mathcal{F}$, and removes any entries that are dominated by the new pair. The test (3.30) provides an important inclusion property that if a pair (h, f) is added to the filter, then the set of points that are unacceptable for the new filter always includes the points that are unacceptable for the old filter.

As in the Burke-Han approach of Section 3.2.3, the principal goal of a filter method is the attainment of feasibility. An important property of the filter defined above is that if there are an infinite sequence of iterations in which $(h(x_k), f(x_k))$ is entered into the filter, and $\{f(x_k)\}$ is bounded below, then $h(x_k) \to 0$ (see Fletcher, Leyffer and Toint [69]).

3.4.1. Trust-region filter methods

Fletcher and Leyffer [67,68] propose a trust-region filter method in which a filter is used to accept or reject points generated by a plain SQP subproblem with a trust-region constraint.

Below we give a brief description of the variant of the Fletcher-Leyffer method proposed by Fletcher, Leyffer and Toint [69]. The filter is defined in terms of the one-norm of the constraint violations, i.e., $h(x) = \|c(x)\|_1$, and the trust-region subproblem is given by

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f_k + g_k^T(x - x_k) + \tfrac{1}{2}(x - x_k)^T H_k(x - x_k) \\ \text{subject to} & c_k + J_k(x - x_k) = 0, \quad x \geq 0, \quad \|(x - x_k)\|_\infty \leq \delta_k. \end{array} \tag{3.31}$$

To simplify the discussion, we start by assuming that the QP subproblem (3.31) remains feasible. In this case, the filter method generates a sequence of points $\{x_k\}$ and a corresponding sequence of filters $\{\mathcal{F}_k\}$ such that x_k is acceptable to the filter $\mathcal{F}_k$ and $x_{k+1} = \widehat{x}_k$, where $\widehat{x}_k$ is a global minimizer of the QP subproblem (3.31). The use of a filter alone does not necessarily enforce convergence to a solution of the constrained problem. For example, if the iterates converge to an arbitrary feasible point and the infeasibility measure h is reduced by a factor of at least β at each iteration, then the iterates will be acceptable to the filter independently of f. This implies that the filter must be used in conjunction with a sufficient reduction condition analogous to (3.1), i.e.,

$$\mathit{\Delta} m_k(d_k) > 0 \quad \text{and} \quad f(x_k) - f(x_k + d_k) \geq \eta \mathit{\Delta} m_k(d_k), \tag{3.32}$$

where $d_k = \widehat{x}_k - x_k$, and $\mathit{\Delta} m_k(d_k) = m_k(x_k) - m_k(x_k + d_k)$ for a local model $m_k(x)$ of f (e.g., $m_k(x) = f(x_k) + g(x_k)^T(x - x_k)$).

At the start of the kth iteration, we have a point x_k and a filter $\mathcal{F}_{k-1}$ such that (h_k, f_k) is acceptable to $\mathcal{F}_{k-1}$, but is not yet included in $\mathcal{F}_{k-1}$ (it will be shown below that x_k may or may not be included in the filter even though it constitutes an acceptable entry). The kth iteration is analogous to that of a backtracking line-search method, except that the backtracking steps are performed by solving the QP (3.31) with decreasing values of the trust-region radius. The backtracking continues until $\widehat{x}_k$ is acceptable to the combined filter $\mathcal{F}_{k-1} \cup (h_k, f_k)$, and either $f(x_k) - f(x_k + d_k) \geq \eta \mathit{\Delta} m_k(d_k)$ or $\mathit{\Delta} m_k(d_k) \leq 0$. On termination of the backtracking procedure, if $\mathit{\Delta} m_k(d_k) \leq 0$, then (h_k, f_k) is added to $\mathcal{F}_{k-1}$ (giving $\mathcal{F}_k$), otherwise $\mathcal{F}_k = \mathcal{F}_{k-1}$. Finally, the next iterate is defined as $x_{k+1} = \widehat{x}_k$ and the trust-region radius δ_{k+1} for the next iteration is initialized at some value greater than some preassigned minimum value $\delta_{\min}$. This reinitialization provides the opportunity to increase the trust-region radius based on the change in f. For example, the trust region radius can be increased if the predicted reduction in f is greater that some positive factor of h.

As mentioned above, although (h_k, f_k) is acceptable to $\mathcal{F}_{k-1}$, it is not necessarily added to the filter. The point x_k is added if and only if $\mathit{\Delta} m_k(d_k) \leq 0$, in which case the QP solution predicts an increase in f, and the primary aim of the iteration changes to that of reducing h (by allowing f to increase if necessary). The requirement that $\mathit{\Delta} m_k(d_k) \leq 0$ for adding to the filter ensures that all the filter entries have $h_j > 0$. This is because if $h_k = 0$, then the QP must be compatible (even without this being an assumption), and hence, if x_k is not a KKT point, then $\mathit{\Delta} m_k(d_k) > 0$ and x_k is not added to the filter.

Now we drop our assumption that the QP problem (3.31) is always feasible. If a new entry is never added to the filter during the backtracking procedure, then $\delta_k \to 0$ and there are two situations that can occur. If $c(x_k) = 0$, then the problem looks like an unconstrained problem. If f is reduced then we must make progress and conventional trust-region theory applies. On the other hand, if $c(x_k) \neq 0$, then reducing the trust-region radius will eventually give an infeasible QP. In this case, the method switches to a *restoration phase* that focuses on minimizing $h(x)$ subject to $x \geq 0$. In this case a *restoration filter* may be defined that allows nonmonotone progress on $h(x)$. Note that it is possible for the QP to be infeasible for any infeasible x_k. In this situation the filter method will converge to a nonoptimal local minimizer of $h(x)$ (just as the Han-Powell method may converge to a nonoptimal local minimizer of the merit function).

The convergence properties of filter-SQP methods are similar to those of methods that use a merit function. In particular, it is possible to establish convergence to either a point that satisfies the first-order necessary conditions for optimality, or a point that minimizes $h(x)$ locally (see Fletcher, Leyffer and Toint [69] for details). It is not necessary that H_k be positive definite, although $\widehat{x}_k$ must be a global solution of the QP subproblem (3.31) (see the cautionary opening remarks of the Appendix concerning the solution of indefinite QPs). Standard examples that exhibit the Maratos effect for an SQP method with a merit function cause no difficulties for the filter method. Although the unit step causes an increase in the constraint violation, and hence an increase in a penalty function, it also causes a decrease in the objective and so it is acceptable to the filter. However, Fletcher and Leyffer [68] give a simple example for which the QP solution increases both the objective and the constraint violations, resulting in a reduction in the trust-region radius and the rejection of the Newton step. Fletcher and Leyffer propose the use of a second-order correction step analogous to (3.27). Ulbrich [173] defines a filter that uses the Lagrangian function instead of f and shows that superlinear convergence may be obtained without using the second-order correction.

3.4.2. Line-search filter methods

The trust-region filter method described in Section 3.4.1 may be modified to use a line search by solving the plain SQP subproblem and replacing the backtracking trust-region procedure by a conventional backtracking line search. In this case, the candidate pair for the filter is $\big(h(x_k + \alpha_k p_k), f(x_k + \alpha_k p_k)\big)$, where α_k is a member of a decreasing sequence of steplengths, and $p_k = \widehat{x}_k - x_k$, with $\widehat{x}_k$ a solution of the plain QP (2.17). Analogous to (3.32), the sufficient decrease criteria for the objective are

$$\varDelta m_k(\alpha_k p_k) > 0 \quad \text{and} \quad f(x_k) - f(x_k + \alpha_k p_k) \geq \eta \varDelta m_k(\alpha_k p_k).$$

If the trial step length is reduced below a minimum value $\alpha_k^{\min}$, the line search is abandoned and the algorithm switches to the restoration phase. For more details, the reader is referred to the two papers of Wächter and Biegler [176, 175]. The caveats of the previous section concerning the definition of H_k also apply to the line-search filter method. In addition, the absence of an explicit bound on $\|x - x_k\|$ provided by the trust-region constraint adds the possibility of unboundedness of the QP subproblem.

Chin, Rashid and Nor [34] consider a line-search filter method that includes a second-order correction step during the backtracking procedure. If $x_k + \alpha p_k$ is not acceptable to the filter, a second-order correction s_k is defined by solving the equality-constrained QP:

$$\begin{array}{ll} \underset{s\in\mathbb{R}^n}{\text{minimize}} & f_k + g_k^T(p_k + s) + \frac{1}{2}(p_k + s)^T H_k (p_k + s) \\ \text{subject to} & c(x_k + p_k) + J_k s = 0, \quad (x_k + p_k + s)_{\mathcal{A}} = -\|p_k\|^\nu e, \end{array} \tag{3.33}$$

where $\nu \in (2, 3)$ and $\mathcal{A}(\widehat{x}_k)$ is the active set predicted by the QP subproblem (for a similar scheme, see Herskovits [123], and Panier and Tits [149, 150]). Given an optimal solution s_k, Chin, Rashid and Nor [34] show that under certain assumptions, the sufficient decrease criteria

$$f(x_k) - f(x_k + \alpha_k p_k + \alpha_k^2 s_k) \geq \eta \varDelta m_k(\alpha_k p_k) \;\text{ and }\; \varDelta m_k(\alpha_k p_k) > 0$$

give a sequence $\{x_k\}$ with local Q-superlinear convergence.

3.5. SQP methods based on successive LP and QP

In the MINLP context, it is necessary to solve a sequence of related nonlinear programs, some with infeasible constraints. For maximum efficiency, it is crucial that the active set from one problem is used to provide a warm start for the next. A substantial benefit of SQP methods is that they are easily adapted to accept an estimate of the active set. However, if warm starts are to be exploited fully, it is necessary that the second derivatives of the problem functions are available and that these derivatives are utilized by the SQP method. Unfortunately, none of the SQP methods discussed in Sections 3.2–3.4 are completely suitable for use with second derivatives. The main difficulty stems from the possibility that the Hessian of the Lagrangian is indefinite, in which case the inequality constrained QP subproblem is nonconvex. A nonconvex QP is likely to have many local solutions, and may be unbounded. Some SQP methods are only well-defined if the subproblem is convex—e.g., methods that rely on the use of a positive-definite quasi-Newton approximate Hessian. Other methods require the calculation of a global solution of the QP subproblem, which has the benefit of ensuring that the "same" local solution is found for the final sequence of related QPs. Unfortunately, nonconvex quadratic programming is NP-hard, and even the seemingly simple task of checking for local optimality is intractable when there are zero Lagrange multipliers (see the opening remarks of the Appendix).

One approach to resolving this difficulty is to estimate the active set using a *convex programming approximation* of the plain QP subproblem (2.1). This active set is then used to define an equality-constrained QP (EQP) subproblem whose solution may be used in conjunction with a merit function or filter to obtain the next iterate. One of the first methods to use a convex program to estimate the active set was proposed by Fletcher and Sainz de la Maza [70], who proposed estimating the active set by solving a linear program with a trust-region constraint. (Their method was formulated first as a sequential unconstrained method for minimizing a nonsmooth composite function. Here we describe the particular form of the method in terms of minimizing the ℓ_1 penalty function $P_1(x,\rho)$ defined in (1.8).) The convex subproblem has the form

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & l_k(x) = f_k + g_k^T(x-x_k) + \rho\|\widehat{c}_k(x)\|_1 + \rho\|\,[\,x\,]_-\,\|_1 \\ \text{subject to} & \|x - x_k\| \le \delta_k, \end{array} \tag{3.34}$$

which involves the minimization of a piecewise linear function subject to a trust-region constraint (cf. (3.26)). If the trust-region constraint is defined in terms of the infinity-norm, the problem (3.34) is equivalent to the linear programming (LP) problem:

$$\begin{array}{ll} \underset{x,w\in\mathbb{R}^n;\,u,v\in\mathbb{R}^m}{\text{minimize}} & f_k + g_k^T(x-x_k) + \rho e^T u + \rho e^T v + \rho e^T w \\ \text{subject to} & c_k + J_k(x-x_k) - u + v = 0, \quad u \ge 0, \quad v \ge 0, \\ & x_k - \delta_k e \le x \le x_k + \delta_k e, \quad x + w \ge 0, \quad w \ge 0. \end{array} \tag{3.35}$$

This equivalence was the motivation for the method to be called the *successive linear programming* (SLP) method. Fletcher and Sainz de la Maza use the reduction in $P_1(x,\rho)$ predicted by the *first-order* subproblem (3.34) to assess the quality of the reduction $P_1(x_k,\rho) - P_1(x_k+d_k,\rho)$ defined by a *second-order* method (to be defined below).

Given a positive-definite approximation B_k of the Hessian of the Lagrangian, let $q_k(x)$ denote the piecewise quadratic function

$$q_k(x) = l_k(x) + \tfrac{1}{2}(x-x_k)^T B_k(x-x_k).$$

Let $d_k^{\text{LP}} = \widehat{x}_k^{\text{LP}} - x_k$, where $\widehat{x}_k^{\text{LP}}$ is a solution of the LP (3.35), and define $\Delta l_k = l_k(x_k) - l_k(x_k + d_k^{\text{LP}})$. Then it holds that

$$q_k(x_k) - \min_d q_k(x_k+d) \ge \tfrac{1}{2}\Delta l_k \min\{\Delta l_k/\beta_k, 1\},$$

where $\beta_k = (d_k^{\text{LP}})^T B_k d_k^{\text{LP}} > 0$. This inequality suggests that a suitable acceptance criterion for an estimate $x_k + d$ is

$$P_1(x_k,\rho) - P_1(x_k+d,\rho) \ge \eta\Delta l_k \min\{\Delta l_k/\beta_k, 1\},$$

where η is some preassigned scalar such that $0 < \eta < \frac{1}{2}$. This criterion is used to determine if the new iterate x_{k+1} should be set to (i) the current iterate x_k (which always triggers a reduction in the trust-region radius); (ii) the second-order step $x_k + d_k$; or (iii) the first-order step $x_k + d_k^{\text{LP}}$. The test for accepting the second-order step is done first, If the second-order step fails, then the penalty function is recomputed at $x_k + d_k^{\text{LP}}$ and the test is repeated to determine if x_{k+1} should be $x_k + d_k^{\text{LP}}$. Finally, the trust-region radius is updated based on a conventional trust-region strategy that compares the reduction in the penalty function with the reduction predicted by the LP subproblem (the reader is referred to the original paper for details).

Next, we consider how to define a second-order step. Let $\mathcal{B}$ and $\mathcal{N}$ denote the final LP basic and nonbasic sets for the LP (3.35). To simplify the description, assume that the optimal u, v and w are zero. A second-order iterate $\widehat{x}_k$ can be defined as the solution of an equality-constrained quadratic program (EQP) defined by minimizing the quadratic model $\widehat{f}_k(x) = f_k + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k)$ subject to $c_k + J_k(x - x_k) = 0$, with the nonbasic variables fixed at their current values. Let $p_k = \widehat{x}_k - x_k$, where $(\widehat{x}_k, \widehat{\pi}_k)$ is the primal-dual EQP solution. Let $\widetilde{p}_k$ denote the vector of components of p_k in the final LP basic set $\mathcal{B}$, with $\widetilde{J}_k$ the corresponding columns of J_k. The vector $(\widetilde{p}_k, \widehat{\pi}_k)$ satisfies the KKT equations

$$\begin{pmatrix} \widetilde{B}_k & \widetilde{J}_k^T \\ \widetilde{J}_k & 0 \end{pmatrix} \begin{pmatrix} \widetilde{p}_k \\ -\widehat{\pi}_k \end{pmatrix} = - \begin{pmatrix} (g_k + B_k \eta_k)_{\mathcal{B}} \\ c_k + J_k \eta_k \end{pmatrix}, \tag{3.36}$$

where η_k is defined in terms of the final LP nonbasic set, i.e.,

$$(\eta_k)_i = \begin{cases} (\widehat{x}_k^{\text{LP}} - x_k)_i & \text{if } i \in \mathcal{N}; \\ 0 & \text{if } i \notin \mathcal{N}. \end{cases}$$

There are many ways of solving these KKT equations. The most appropriate method will depend on certain basic properties of the problem being solved, which include the size of the problem (i.e., the number of variables and constraints); whether or not the Jacobian is dense or sparse; and how the approximate Hessian is stored (see, e.g., Section 3.2.1). Fletcher and Sainz de la Maza suggest finding an approximate solution of the EQP using a quasi-Newton approximation of the reduced Hessian matrix (see Coleman and Conn [37]).

The results of Fletcher and Sainz de la Maza may be used to show that, under reasonable nondegeneracy and second-order conditions, the active set of the LP subproblem (3.35) ultimately predicts that of the smooth variant of the penalty function at limit points of $\{x_k\}$. This implies fast asymptotic convergence. Fletcher and Sainz de la Maza did not consider the use of exact second derivatives in their original paper, and it took more than 12 years before the advent of reliable second-derivative trust-region and filter methods for the EQP subproblem allowed the potential of SLP methods to be realized. Chin and Fletcher [33] proposed the use of a trust-region filter method that does not require the use of the ℓ_1 penalty function. For a similar approach that uses a filter, see Fletcher et al. [66]. In a series of papers, Byrd, Gould, Nocedal and Waltz [26, 27] proposed a method that employs an additional trust region to safeguard the EQP direction. They also define an appropriate method for adjusting the penalty parameter. Recently, Morales, Nocedal and Wu [139], and Gould and Robinson [107, 108, 109] have proposed SQP methods that identify the active set using a convex QP based on a positive-definite BFGS approximation of the Hessian.

4. SQP issues relevant to MINLP

4.1. Treatment of linear constraints

An important feature of SQP methods is that it is relatively easy to exploit the special properties of linear constraints. This can be an advantage when a method for MINLP solves a sequence of NLPs that differ by only the number of linear constraints that are imposed. Suppose that the general linear constraints are a subset of the constraints defined by $c(x) = 0$, e.g., $c_L(x) = Ax - b = 0$. Then a feasible point for the linear constraints $c_L(x) = Ax - b = 0$, $x \geq 0$, can be found by solving the elastic problem

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n;\, u,v\in\mathbb{R}^m}{\text{minimize}} & \rho e^T u + e^T v \\ \text{subject to} & Ax - u + v = b, \quad x \geq 0, \quad u \geq 0, \quad v \geq 0. \end{array} \tag{4.1}$$

This is equivalent to minimizing the one-norm of the general linear constraint violations subject to the simple bounds. An important property of linear constraints is that it is possible to determine the solvability of a system of linear inequalities in a finite number of steps. If the linear constraints are infeasible ($u \neq 0$ or $v \neq 0$), then the SQP algorithm can terminate without computing the nonlinear functions. Otherwise, all subsequent iterates satisfy the linear constraints.

4.2. Treatment of infeasibilities

If the constraints of the QP subproblem (2.1) have no feasible point, then no QP solution exists. This could be for two reasons, either: (i) the NLP is feasible but the quadratic programming subproblem is locally infeasible, or (ii) the NLP is infeasible. If the NLP is convex, then infeasibility of the quadratic programming subproblem implies infeasibility of the original problem, but in the nonconvex case, there is no such implication.

If the subproblem is infeasible, the algorithm may continue in *elastic mode*, by solving the elastic QP (3.22). There are two interpretations of the role of the elastic QP. In one interpretation, the elastic problem defines a regularization of the plain QP subproblem (2.1). In this case, if the NLP is feasible and $\rho_k \geq \|\pi_{k+1}\|_\infty$, then problems (2.1) and (3.22) are equivalent. An alternative interpretation is to view the elastic QP as the QP subproblem associated with the elastic *nonlinear* problem (1.4), so that the elastic constraints are present in the original problem and are inherited by the QP subproblem. Note that any solution of the NLP may be regarded as a solution of (3.22) for a value of ρ_k such that $\rho_k \geq \|\pi_{k+1}\|_\infty$. Hence, even if ρ_k is not present explicitly, we may consider both the subproblem (3.22) and the original problem (1.4) to have an associated implicit value of ρ_k that is larger than $\|\pi_{k+1}\|_\infty$.

4.3. Infeasibility detection

As we discussed in Section 1.3, it is important to be able to determine as quickly as possible if the NLP (1.2) is infeasible. In an SQP framework, infeasibility may be detected either by solving the QP subproblem in elastic mode (3.22) with a sequence of penalty parameters $\rho_k \to \infty$, or by solving a sequence of elastic *nonlinear* problems of the form (1.4) with $\rho_k \to \infty$. For an SQP method that solves a sequence of nonlinear elastic problems and uses a *quasi-Newton approximation* to the Hessian, infeasibility is usually signaled by the occurrence of a sequence of elastic problems in which the penalty parameter is increased, but the current x_k remains fixed, i.e., an optimal solution for a problem with $\rho = \rho_k$ is optimal for the problem with $\rho = \rho_{k+1}$, etc. This is usually a reliable indication that x_k is a local minimizer of the sum of infeasibilities. This behavior can be explained by the fact that a warm start uses the approximate Hessian from the previous elastic problem, which is not changed as ρ_k and the QP-multipliers are increased. This is one situation where the inability of a quasi-Newton Hessian to adapt to changes in the multipliers is beneficial!

The situation is different when the SQP method uses the exact Hessian of the Lagrangian. In this case, the multipliers reflect the magnitude of ρ_k, and so the Hessian changes substantially. In the following, we give a brief discussion of this case that reflects the paper of

Byrd, Curtis and Nocedal [25]. For an infeasible problem, it must hold that $\rho_k \to \infty$ and $\rho_k > \rho_{k-1}$ for an infinite subsequence of iterates. In this situation, *different* problems are being solved at outer iterations $k-1$ and k. At iteration $k-1$, the problem is the elastic problem (1.4) with $\rho = \rho_{k-1}$, whereas at iteration k, the problem is the elastic problem with $\rho = \rho_k$. We may write

$$f(x) + \rho_k e^T u + \rho_k e^T v = f(x) + \frac{\rho_k}{\rho_{k-1}}\big(\rho_{k-1} e^T u + \rho_{k-1} e^T v\big). \tag{4.2}$$

If the NLP is infeasible, it must hold that $\|u\| + \|v\| > 0$. If ρ_{k-1} is large, with $\rho_k > \rho_{k-1}$ and $\|u\| + \|v\| > 0$, then the term $f(x)$ is negligible in (4.2), i.e., $f(x) \ll \rho_{k-1} e^T u + \rho_{k-1} e^T v$, so that

$$\begin{aligned} f(x) + \rho_k e^T u + \rho_k e^T v &\approx \rho_k e^T u + \rho_k e^T v \\ &= \frac{\rho_k}{\rho_{k-1}}\left(\rho_{k-1} e^T u + \rho_{k-1} e^T v\right) \\ &\approx \frac{\rho_k}{\rho_{k-1}}\left(f(x) + \rho_{k-1} e^T u + \rho_{k-1} e^T v\right). \end{aligned} \tag{4.3}$$

The form of (4.3) implies that the elastic problems at iterations $k-1$ and k differ (approximately) by only a multiplicative factor ρ_k/ρ_{k-1} in the scaling of the objective function. The approximation becomes increasingly accurate as ρ_{k-1} tends to infinity.

Let (x_k, u_k, v_k) be the solution provided by the elastic QP subproblem at iteration $k-1$, with corresponding Lagrange multiplier estimates (π_k, z_k). Also assume that (x_k, u_k, v_k) is close to optimal for the corresponding elastic problem (1.4) with $\rho = \rho_{k-1}$. If $\rho_k > \rho_{k-1}$, the question is how to provide a good initial point to this new problem. If (x_k, u_k, v_k) is the exact solution of the elastic problem for $\rho = \rho_{k-1}$, then (π_k, z_k) are the corresponding Lagrange multipliers. Moreover, if the objective functions differ by the factor ρ_k/ρ_{k-1}, then (x_k, u_k, v_k) is again optimal for the new problem, and the dual variables inherit the same scaling as the objective function (see (1.6b)). In this situation, the new multipliers are $\big((\rho_k/\rho_{k-1})\pi_k, (\rho_k/\rho_{k-1})z_k\big)$. Based on these observations, in an idealized situation, we expect that (x_k, u_k, v_k), together with scaled Lagrange multiplier estimates $(\rho_k/\rho_{k-1})\pi_k$ and $(\rho_k/\rho_{k-1})z_k$, provide good initial estimates for the new elastic QP subproblem. Hence, if second derivatives are used in the QP subproblem, the Hessian of the Lagrangian should be evaluated at (x_k, u_k, v_k) with Lagrange multiplier estimates $\big((\rho_k/\rho_{k-1})\pi_k, (\rho_k/\rho_{k-1})z_k\big)$ in order to obtain fast convergence as ρ_k increases.

As ρ_k tends to infinity, the objective function becomes less important compared to the penalty term in the objective of (1.4). Eventually only the infeasibilities matter, and the iterates converge to a local minimizer of the sum of infeasibilities. See Byrd, Curtis and Nocedal [25] for a detailed discussion on infeasibility detection, including a discussion on how to let $\rho_k \to \infty$ rapidly.

4.4. Solving a sequence of related QP subproblems

In MINLP branch and bound methods it is necessary to solve a sequence of NLPs that differ by a single constraint (see, e.g., Leyffer [130], and Goux and Leyffer [113]). For example, at the solution of a relaxed problem, some integer variables take a non-integer value. The MINLP algorithm selects one of the integer variables that takes a non-integer value, say x_i with value $\bar{x}_i$, and branches on it. Branching generates two new NLP problems by adding simple bounds $x_i \le \lfloor \bar{x}_i \rfloor$ and $x_i \ge \lfloor \bar{x}_i \rfloor + 1$ to the NLP relaxation (where $\lfloor v \rfloor$ is the largest integer not greater than v). The SQP methods of Section 3.5 that solve an initial convex programming problem to determine the active set have the advantage that a dual QP/LP solver may be used to solve the convex QP subproblem (dual active-set QP methods are discussed in Section A.2). This provides similar advantages to MINLP solvers as the dual simplex method provides to MILP. If the SQP method is implemented with a dual QP solver, and is warm started with the primal-dual solution of the previous relaxation, then the dual variables are feasible and only one branched variable is infeasible. The infeasible x_i can be moved towards feasibility immediately.

A similar situation applies if a nonlinear cut adds a constraint to the NLP. For simplicity, assume that the QP has objective $g^Tx + \frac{1}{2}x^THx$ and constraints $Ax = b$, $x \geq 0$. As the QP is in standard form, the cut adds a new row and column to A, a zero element to the objective g, and a zero row and column to H. This gives a new problem with $\bar{A}$, $\bar{b}$, $\bar{g}$ and $\bar{H}$ (say). The new column of $\bar{A}$ corresponds to the unit vector associated with the new slack variable. An obvious initial basis for the new problem is

$$\bar{A}_B = \begin{pmatrix} A_B & 0 \\ a^T & 1 \end{pmatrix},$$

so the new basic solution $\bar{x}_B$ is the old solution x_B, augmented by the new slack, which is infeasible. This means that if we solve the primal QP then it would be necessary to go into phase 1 to get started. However, by solving the dual QP, then we have an initial feasible subspace minimizer for the dual based on a $\bar{y}_B (= \bar{x}_B)$ such that $\bar{A}_B\bar{y}_B = \bar{b}$ and

$$\bar{z} = \bar{g} + \bar{H}\bar{y} - \bar{A}^T\bar{\pi}.$$

We can choose $\bar{\pi}$ to be the old π augmented by a zero. The new element of $\bar{y}_B$ corresponds to the new slack, so the new elements of $\bar{g}$ and row and column of $\bar{H}$ are zero. This implies that $\bar{z}$ is essentially z, and hence $\bar{z} \geq 0$.

Acknowledgments

The authors would like to thank Anders Forsgren for numerous discussions on SQP methods during the preparation of this paper. We are also grateful to the referees for suggestions that substantially improved the presentation.

APPENDIX

A. Methods for Quadratic Programming

We consider methods for the quadratic program

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & g^T(x-x_I)+\tfrac{1}{2}(x-x_I)^TH(x-x_I) \\ \text{subject to} & Ax = Ax_I - b, \quad x\ge 0, \end{array} \tag{A.1}$$

where g, H, b, A and x_I are given constant quantities, with H symmetric. The QP objective is denoted by $\widehat{f}(x)$, with gradient $\widehat{g}(x) = g + H(x-x_I)$. In some situations, the general constraints will be written as $\widehat{c}(x)=0$, with $\widehat{c}(x) = A(x-x_I)+b$. The QP active set is denoted by $\mathcal{A}(x)$. A primal-dual QP solution is denoted by (x^*,π^*,z^*). In terms of the QP defined at the kth outer iteration of an SQP method, we have $x_I = x_k$, $b = c(x_k)$, $g = g(x_k)$, $A = J(x_k)$ and $H = H(x_k,\pi_k)$. It is assumed that A has rank m. No assumptions are made about H other than symmetry. Conditions that must hold at an optimal solution of (A.1) are provided by the following result (see, e.g., Borwein [12], Contesse [44] and Majthay [134]).

Result A.1. (QP optimality conditions)
The point x^* is a local minimizer of the quadratic program (A.1) if and only if

(a) *$\widehat{c}(x^*)=0$, $x^*\ge 0$, and there exists at least one pair of vectors π^* and z^* such that $\widehat{g}(x^*) - A^T\pi^* - z^* = 0$, with $z^*\ge 0$, and $z^*\cdot x^* = 0$;*

(b) *$p^THp \ge 0$ for all nonzero p satisfying $\widehat{g}(x^*)^Tp = 0$, $Ap = 0$, and $p_i\ge 0$ for every $i\in\mathcal{A}(x^*)$.* ∎

Part (a) gives the first-order KKT conditions (2.18) for the QP (A.1). If H is positive semidefinite, the first-order KKT conditions are both necessary and sufficient for (x^*,π^*,z^*) to be a local primal-dual solution of (A.1).

Suppose that (x^*,π^*,z^*) satisfies condition (a) with $z_i^*=0$ and $x_i^*=0$ for some i. If H is positive semidefinite, then x^* is a *weak minimizer* of (A.1). In this case, x^* is a global minimizer with a unique global minimum $\widehat{f}(x^*)$. If H has at least one negative eigenvalue, then x^* is known as a *dead point*. Verifying condition (b) at a dead point requires finding the global minimizer of an indefinite quadratic form over a cone, which is an NP-hard problem (see, e.g., Cottle, Habetler and Lemke [45], Pardalos and Schnitger [151], and Pardalos and Vavasis [152]). This implies that the optimality of a candidate solution of a general quadratic program can be verified only if more restrictive (but computationally tractable) sufficient conditions are satisfied. A dead point is a point at which the sufficient conditions are not satisfied, but certain necessary conditions hold. Computationally tractable necessary conditions are based on the following result.

Result A.2. (Necessary conditions for optimality)
The point x^* is a local minimizer of the QP (A.1) only if

(a) *$\widehat{c}(x^*)=0$, $x^*\ge 0$, and there exists at least one pair of vectors π^* and z^* such that $\widehat{g}(x^*) - A^T\pi^* - z^* = 0$, with $z^*\ge 0$, and $z^*\cdot x^* = 0$;*

(b) *$p^THp\ge 0$ for all nonzero p satisfying $Ap=0$, and $p_i = 0$ for every $i\in\mathcal{A}(x^*)$.* ∎

Suitable sufficient conditions for optimality are given by (a)–(b) with (b) replaced by the condition that $p^THp \ge \omega\|p\|^2$ for some $\omega > 0$ and all p such that $Ap = 0$, and $p_i = 0$ for every $i \in \mathcal{A}_+(x)$, where $\mathcal{A}_+(x)$ is the index set $\mathcal{A}_+(x) = \{i \in \mathcal{A}(x) : z_i > 0\}$.

Typically, software for general quadratic programming is designed to terminate at a dead point. Nevertheless, it is possible to define procedures that check for optimality at a dead point, but the chance of success in a reasonable amount of computation time depends on the dimension of the problem (see Forsgren, Gill and Murray [73]).

A.1. Primal active-set methods

We start by reviewing the properties of *primal-feasible active-set methods* for quadratic programming. An important feature of these methods is that once a feasible iterate is found, all subsequent iterates are feasible. The methods have two phases. In the first phase (called the *feasibility phase* or *phase one*), a feasible point is found by minimizing the sum of infeasibilities. In the second phase (the *optimality phase* or *phase two*), the quadratic objective function is minimized while feasibility is maintained. Each phase generates a sequence of inner iterates $\{x_j\}$ such that $x_j \ge 0$. The new iterate x_{j+1} is defined as $x_{j+1} = x_j + \alpha_j p_j$, where the *step length* α_j is a nonnegative scalar, and p_j is the *QP search direction.* For efficiency, it is beneficial if the computations in both phases are performed by the same underlying method. The two-phase nature of the algorithm is reflected by changing the function being minimized from a function that reflects the degree of infeasibility to the quadratic objective function. For this reason, it is helpful to consider methods for the optimality phase first.

At the jth step of the optimality phase, $\widehat{c}(x_j) = A(x_j - x_I) + b = 0$ and $x_j \ge 0$. The vector p_j is chosen to satisfy certain properties with respect to the objective and constraints. First, p_j must be a *direction of decrease* for $\widehat{f}$ at x_j, i.e., there must exist a positive $\bar{\alpha}$ such that

$$\widehat{f}(x_j + \alpha p_j) < \widehat{f}(x_j) \quad \text{for all } \alpha \in (0, \bar{\alpha}].$$

In addition, $x_j + p_j$ must be feasible with respect to the general constraints, and feasible with respect to the bounds associated with a certain "working set" of variables that serves as an estimate of the optimal active set of the QP. Using the terminology of linear programming, we call this working set of variables the *nonbasic set*, denoted by $\mathcal{N} = \{\nu_1, \nu_2, \ldots, \nu_{n_N}\}$. Similarly, we define the set $\mathcal{B}$ of indices that are not in $\mathcal{N}$ as the *basic set*, with $\mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_{n_B}\}$, where $n_B = n - n_N$. Although $\mathcal{B}$ and $\mathcal{N}$ are strictly index sets, we will follow common practice and refer to variables x_{β_r} and x_{ν_s} as being "in $\mathcal{B}$" and "in $\mathcal{N}$" respectively.

With these definitions, we define the columns of A indexed by $\mathcal{N}$ and $\mathcal{B}$, the *nonbasic* and *basic* columns of A, as A_N and A_B, respectively. We refrain from referring to the nonbasic and basic sets as the "fixed" and "free" variables because some active-set methods allow some nonbasic variables to move (the simplex method for linear programming being one prominent example). An important attribute of the nonbasic set is that A_B *has rank* m, i.e., the rows of A_B are linearly independent. This implies that the cardinality of the nonbasic set must satisfy $0 \le n_N \le n - m$. It must be emphasized that our definition of $\mathcal{N}$ does not require a nonbasic variable to be active (i.e., at its lower bound). Also, whereas the active set is defined uniquely at each point, there are many choices for $\mathcal{N}$ (including the empty set). Given any n-vector y, the vector of *basic components of* y, denoted by y_B, is the n_B-vector whose jth component is component β_j of y. Similarly, y_N, the vector *nonbasic components of* y, is the n_N-vector whose jth component is component ν_j of y.

Given a basic-nonbasic partition of the variables, we introduce the definitions of stationarity and optimality with respect to a basic set.

Definition A.1. (Subspace stationary point) *Let $\mathcal{B}$ be a basic set defined at an $\widehat{x}$ such that $\widehat{c}(\widehat{x}) = 0$. Then $\widehat{x}$ is a* subspace stationary point with respect to $\mathcal{B}$ (or, equivalently, with respect to A_B) *if there exists a vector π such that $\widehat{g}_B(\widehat{x}) = A_B^T\pi$. Equivalently, $\widehat{x}$ is a subspace stationary point with respect to $\mathcal{B}$ if the reduced gradient $Z_B^T\widehat{g}_B(\widehat{x})$ is zero, where the columns of Z_B form a basis for the null-space of A_B.* ∎

If $\widehat{x}$ is a subspace stationary point, $\widehat{f}$ is stationary on the subspace $\{x : A(x - \widehat{x}) = 0,\ x_N = \widehat{x}_N\}$. At a subspace stationary point, it holds that $g(\widehat{x}) = A^T\pi + z$, where $z_i = 0$ for $i \in \mathcal{B}$—i.e., $z_B = 0$. Subspace stationary points may be classified based on the curvature of $\widehat{f}$ on the nonbasic set.

Definition A.2. (Subspace minimizer) *Let $\widehat{x}$ be a subspace stationary point with respect to $\mathcal{B}$. Let the columns of Z_B form a basis for the null-space of A_B. Then $\widehat{x}$ is a* subspace minimizer with respect to $\mathcal{B}$ *if the reduced Hessian $Z_B^THZ_B$ is positive definite.* ∎

If the nonbasic variables are active at $\widehat{x}$, then $\widehat{x}$ is called a *standard* subspace minimizer. At a standard subspace minimizer, if $z_N \ge 0$ then $\widehat{x}$ satisfies the necessary conditions for optimality. Otherwise, there exists an index $\nu_s \in \mathcal{N}$ such that $z_{\nu_s} < 0$. If some nonbasic variables are not active at $\widehat{x}$, then $\widehat{x}$ is called a *nonstandard* subspace minimizer.

It is convenient sometimes to be able to characterize the curvature of $\widehat{f}$ in a form that does not require the matrix Z_B explicitly. The *inertia* of a symmetric matrix X, denoted by $\mathrm{In}(X)$, is the integer triple (i_+, i_-, i_0), where i_+, i_- and i_0 denote the number of positive, negative and zero eigenvalues of X. Gould [103] shows that if A_B has rank m and $A_BZ_B = 0$, then $Z_B^TH_BZ_B$ is positive definite if and only if

$$\mathrm{In}(K_B) = (n_B, m, 0), \quad \text{where} \quad K_B = \begin{pmatrix} H_B & A_B^T \\ A_B & 0 \end{pmatrix} \tag{A.2}$$

(see Forsgren [71] for a more general discussion, including the case where A_B does not have rank m). Many algorithms for solving symmetric equations that compute an explicit matrix factorization of K_B also provide the inertia as a by-product of the calculation, see, e.g., Bunch [18], and Bunch and Kaufman [19].

Below, we discuss two alternative formulations of an active-set method. Each generates a feasible sequence $\{x_j\}$ such that $x_{j+1} = x_j + \alpha_j p_j$ with $\widehat{f}(x_{j+1}) \le \widehat{f}(x_j)$. Neither method requires the QP to be convex, i.e., H need not be positive semidefinite. The direction p_j is defined as the solution of an QP subproblem with equality constraints. Broadly speaking, the nonbasic components of p_j are *specified* and the basic components of p_j are adjusted to satisfy the general constraints $A(x_j + p_j) = Ax_I - b$. If p_B and p_N denote the basic and nonbasic components of p_j, then the nonbasic components are fixed by enforcing constraints of the form $p_N = d_N$, where d_N is a constant vector that characterizes the active-set method being used. The restrictions on p_j define constraints $Ap = 0$ and $p_N = d_N$. Any remaining degrees of freedom are used to define p_j as the direction that produces the largest reduction in $\widehat{f}$. This gives the equality constrained QP subproblem

$$\underset{p}{\text{minimize}} \ \ \widehat{g}(x_j)^Tp + \tfrac{1}{2}p^THp \quad \text{subject to} \quad Ap = 0, \quad p_N = d_N.$$

In the following sections we define two methods based on alternative definitions of d_N. Both methods exploit the properties of a subspace minimizer (see Definition A.2) in order to simplify the linear systems that must be solved.

A.1.1. Nonbinding-direction methods

We start with a method that defines a change in the basic-nonbasic partition at every iteration. In particular, one of three changes occurs: (i) a variable is moved from the basic set to the nonbasic set; (ii) a variable is moved from the nonbasic set to the basic set; or (ii) a variable in the basic set is swapped with a variable in the nonbasic set. These changes result in a column being added, deleted or swapped in the matrix A_B.

In order to simplify the notation, we drop the subscript j and consider the definition of a single iteration that starts at the primal-dual point (x, π) and defines a new iterate $(\bar{x}, \bar{\pi})$ such that $\bar{x} = x + \alpha p$ and $\bar{\pi} = \pi + \alpha q_\pi$. A crucial assumption about (x, π) is that it is a subspace minimizer with respect to the basis $\mathcal{B}$. It will be shown that this assumption guarantees that the next iterate $(\bar{x}, \bar{\pi})$ (and hence each subsequent iterate) is also a subspace minimizer.

Suppose that the reduced cost associated with the sth nonbasic variable is negative, i.e., $z_{\nu_s} < 0$. The direction p is defined so that all the nonbasic components are fixed except for the sth, which undergoes a unit change. This definition implies that a positive step along p increases x_{ν_s} but leaves all the other nonbasics unchanged. The required direction is defined by the equality constrained QP subproblem:

$$\underset{p}{\text{minimize}} \;\; \widehat{g}(x)^T p + \tfrac{1}{2} p^T H p \quad \text{subject to} \quad Ap = 0, \quad p_N = e_s, \tag{A.3}$$

and is said to be *nonbinding* with respect to the nonbasic variables. If the multipliers for the constraints $Ap = 0$ are defined in terms of an increment q_π to π, then p_B and q_π satisfy the optimality conditions

$$\left(\begin{array}{cc|c} H_B & -A_B^T & H_D \\ A_B & 0 & A_N \\ \hline 0 & 0 & I_N \end{array}\right) \begin{pmatrix} p_B \\ q_\pi \\ p_N \end{pmatrix} = - \begin{pmatrix} \widehat{g}_B(x) - A_B^T \pi \\ 0 \\ -\, e_s \end{pmatrix},$$

where, as above, $\widehat{g}_B(x)$ are the basic components of $\widehat{g}(x)$, and H_B and H_D are the basic rows of the basic and nonbasic columns of H. If x is a subspace minimizer, then $\widehat{g}_B(x) - A_B^T \pi = 0$, so that this system simplifies to

$$\left(\begin{array}{cc|c} H_B & -A_B^T & H_D \\ A_B & 0 & A_N \\ \hline 0 & 0 & I_N \end{array}\right) \begin{pmatrix} p_B \\ q_\pi \\ p_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ e_s \end{pmatrix}, \tag{A.4}$$

yielding p_B and q_π as the solution of the smaller system

$$\begin{pmatrix} H_B & -A_B^T \\ A_B & 0 \end{pmatrix} \begin{pmatrix} p_B \\ q_\pi \end{pmatrix} = - \begin{pmatrix} (h_{\nu_s})_B \\ a_{\nu_s} \end{pmatrix}. \tag{A.5}$$

The increment q_N for multipliers z_N are computed from p_B, p_N and q_π as $q_N = (Hp - A^T q_\pi)_\mathcal{N}$. Once p_B and q_π are known, a nonnegative step α is computed so that $x + \alpha p$ is feasible and $\widehat{f}(x + \alpha p) \le \widehat{f}(x)$. The step that minimizes $\widehat{f}$ as a function of α is given by

$$\alpha_* = \begin{cases} -\widehat{g}(x)^T p / p^T H p & \text{if } p^T H p > 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{A.6}$$

The best feasible step is then $\alpha = \min\{\alpha_*, \alpha_M\}$, where α_M is the maximum feasible step:

$$\alpha_M = \min_{1 \le i \le n_B} \{\gamma_i\}, \quad \text{where} \quad \gamma_i = \begin{cases} \dfrac{(x_B)_i}{-(p_B)_i} & \text{if } (p_B)_i < 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{A.7}$$

(As $p_N = e_s$ and the problem contains only lower bounds, $x+tp$ remains feasible with respect to the nonbasic variables for all $t \ge 0$.) If $\alpha = +\infty$ then $\widehat{f}$ decreases without limit along p and the problem is unbounded. Otherwise, the new iterate is $(\bar{x}, \bar{\pi}) = (x + \alpha p, \pi + \alpha q_\pi)$.

It is instructive to define the step α_* of (A.6) in terms of the identities

$$\widehat{g}(x)^T p = z_{\nu_s} \quad \text{and} \quad p^T H p = (q_N)_s, \tag{A.8}$$

which follow from the equations (A.4) that define p_B and p_N. Then, if α_* is bounded, we have $\alpha_* = -z_{\nu_s}/(q_N)_s$, or, equivalently,

$$z_{\nu_s} + \alpha_*(q_N)_s = 0.$$

Let $z(t)$ denote the vector of reduced costs at any point on the ray $(x + tp, \pi + tq_\pi)$, i.e., $z(t) = \widehat{g}(x + tp) - A^T(\pi + tq_\pi)$. It follows from the definition of p and q_π of (A.4) that $z_B(t) = 0$ for all t, which implies that $x + tp$ is a subspace stationary point *for any step* t. (Moreover, $x + tp$ is a subspace minimizer because the KKT matrix K_B is independent of t.) This property, known as the *parallel subspace property of quadratic programming*, implies that $x + tp$ is the solution of an equality-constraint QP in which the bound on the sth nonbasic is *shifted* to pass through $x + tp$. The component $z_{\nu_s}(t)$ is the reduced cost associated with the shifted version of the bound $x_{\nu_s} \ge 0$. By definition, the sth nonbasic reduced cost is negative at x, i.e., $z_{\nu_s}(0) < 0$. Moreover, a simple calculation shows that $z_{\nu_s}(t)$ is an increasing linear function of t with $z_{\nu_s}(\alpha_*) = 0$ if α_* is bounded. A zero reduced cost at $t = \alpha_*$ means that the shifted bound can be removed from the equality-constraint problem (A.3) (defined at $x = \bar{x}$) without changing its minimizer. Hence, if $\bar{x} = x + \alpha_* p$, the index ν_s is moved to the basic set, which adds column a_{ν_s} to A_B for the next iteration. The shifted variable has been removed from the nonbasic set, which implies that $(\bar{x}, \bar{\pi})$ is a *standard* subspace minimizer.

If we take a shorter step to the boundary of the feasible region, i.e., $\alpha_M < \alpha_*$, then at least one basic variable lies on its bound at $\bar{x} = x + \alpha p$, and one of these, x_{β_r} say, is made nonbasic. If $\bar{A}_B$ denotes the matrix A_B with column r deleted, then $\bar{A}_B$ is not guaranteed to have full row rank (for example, if x is a vertex, A_B is square and $\bar{A}_B$ has more rows than columns). The linear independence of the rows of $\bar{A}_B$ is characterized by the so-called "singularity vector" u_B given by the solution of the equations

$$\begin{pmatrix} H_B & -A_B^T \\ A_B & 0 \end{pmatrix} \begin{pmatrix} u_B \\ v_\pi \end{pmatrix} = \begin{pmatrix} e_r \\ 0 \end{pmatrix}. \tag{A.9}$$

The matrix $\bar{A}_B$ has full rank if and only if $u_B \ne 0$. If $\bar{A}_B$ is rank deficient, $\bar{x}$ is a subspace minimizer with respect to the basis defined by removing x_{ν_s}, i.e., x_{ν_s} is effectively replaced by x_{β_r} in the nonbasic set. In this case, it is necessary to update the dual variables again to reflect the change of basis (see Gill and Wong [98] for more details). The new multipliers are $\bar{\pi} + \sigma v_\pi$, where $\sigma = \widehat{g}(\bar{x})^T p/(p_B)_r$.

As defined above, this method requires the solution of two KKT systems at each step (i.e., equations (A.5) and (A.9)). However, if the solution of (A.9) is such that $u_B \ne 0$, then the vectors p_B and q_π needed at $\bar{x}$ can be updated in $O(n)$ operations using the vectors u_B and v_π. Hence, it is unnecessary to solve (A.5) when a basic variable is removed from $\mathcal{B}$ following a restricted step.

Given an initial standard subspace minimizer x_0 and basic set $\mathcal{B}_0$, this procedure generates a sequence of primal-dual iterates $\{(x_j, \pi_j)\}$ and an associated sequence of basic sets $\{\mathcal{B}_j\}$. The iterates occur in groups of consecutive iterates that start and end at a standard subspace minimizer. Each of the intermediate iterates is a nonstandard subspace minimizer at which the same nonbasic variable may not be on its bound. At each intermediate iterate, a variable moves from $\mathcal{B}$ to $\mathcal{N}$. At the first (standard) subspace minimizer of the group, a nonbasic variable with a negative reduced cost is targeted for inclusion in the basic set. In the subsequent set of iterations, this reduced cost is nondecreasing and the number of basic variables decreases. The group of consecutive iterates ends when the targeted reduced cost reaches zero, at which point the associated variable is made basic.

The method outlined above is based on a method first defined for constraints in all-inequality form by Fletcher [61], and extended to sparse QP by Gould [105]. Recent refinements, including the technique for reducing the number of KKT solves, are given by Gill and Wong [98]. Each of these methods is an example of an *inertia-controlling method*. The idea of an inertia-controlling method is to use the active-set strategy to limit the number of zero and negative eigenvalues in the KKT matrix K_B so that it has inertia $(n_B, m, 0)$ (for a survey, see Gill et al. [94]). At an arbitrary feasible point, a subspace minimizer can be defined by making sufficiently many variables temporarily nonbasic at their current value (see, e.g., Gill, Murray and Saunders [85] for more details).

A.1.2. Binding-direction methods

The next method employs a more conventional active-set strategy in which the nonbasic variables are always active. We start by assuming that the QP is *strictly convex*, i.e., that H is positive definite. Suppose that (x, π) is a feasible primal-dual pair such that $x_i = 0$ for $i \in \mathcal{N}$, where $\mathcal{N}$ is chosen so that A_B has rank m. As in a nonbinding direction method, the primal-dual direction (p, q_π) is computed from an equality constrained QP subproblem. However, in this case the constraints of the subproblem not only force $Ap = 0$ but also require that *every* nonbasic variable remains unchanged for steps of the form $x + \alpha p$. This is done by fixing the nonbasic components of p at zero, giving the equality constraints $Ap = A_B p_B + A_N p_N = 0$ and $p_N = 0$. The resulting subproblem defines a direction that is *binding*, in the sense that it is "bound" or "attached" to the constraints in the nonbasic set. The QP subproblem that gives the best improvement in $\widehat{f}$ is then

$$\underset{p}{\text{minimize}} \;\; \widehat{g}(x)^T p + \tfrac{1}{2} p^T H p \quad \text{subject to} \quad A_B p_B = 0, \;\; p_N = 0. \tag{A.10}$$

The optimality conditions imply that p_B and q_π satisfy the KKT system

$$\begin{pmatrix} H_B & -A_B^T \\ A_B & 0 \end{pmatrix} \begin{pmatrix} p_B \\ q_\pi \end{pmatrix} = - \begin{pmatrix} \widehat{g}_B(x) - A_B^T \pi \\ 0 \end{pmatrix}. \tag{A.11}$$

These equations are nonsingular under our assumptions that H is positive definite and A_B has rank m. If (x, π) is a subspace stationary point, then $z_B = \widehat{g}_B(x) - A_B^T \pi = 0$ and the solution (p_B, q_π) is zero. In this case, no improvement can be made in $\widehat{f}$ along directions in the null-space of A_B. If the components of $z = \widehat{g}(x) - A^T \pi$ are nonnegative then x is optimal for (A.1). Otherwise, a nonbasic variable with a negative reduced cost is selected and moved to the basic set (with no change to x), thereby defining (A.11) with new A_B, H_B and (necessarily nonzero) right-hand side. Given a nonzero solution of (A.11), $x + p$ is either feasible or infeasible with respect to the bounds. If $x + p$ is infeasible, $\mathcal{N}$ cannot be the correct nonbasic set and feasibility is maintained by limiting the step by the maximum feasible step α_M as in (A.7). At the point $\bar{x} = x + \alpha p$, at least one of the basic variables must reach its bound and it is moved to the nonbasic set for the next iteration. Alternatively, if $x + p$ is feasible, $\bar{x} = x + p$ is a subspace minimizer and a nonoptimal nonbasic variable is made basic as above.

The method described above defines groups of consecutive iterates that start with a variable being made basic. No more variables are made basic until either an unconstrained step is taken (i.e., $\alpha = 1$), or a sequence of constrained steps results in the definition of a subspace minimizer (e.g., at a vertex). At each constrained step, the number of basic variables decreases.

As H is positive definite in the strictly convex case, the KKT equations (A.11) remain nonsingular as long as A_B has rank m. One of the most important properties of a binding-direction method is that once an initial nonbasic set is chosen (with the implicit requirement that the associated A_B has rank m), then all subsequent A_B will have rank m (and hence the solution of the KKT system is always well defined). This result is of sufficient importance that we provide a brief proof.

If a variable becomes basic, a column is added to A_B and the rank does not change. It follows that the only possibility for A_B to lose rank is when a basic variable is made nonbasic. Assume that A_B has rank m and that the *first* basic variable is selected to become nonbasic, i.e., $r = 1$. If $\bar{A}_B$ denotes the matrix A_B without its first column, then $A_B = \begin{pmatrix} a_{\beta_r} & \bar{A}_B \end{pmatrix}$. If $\bar{A}_B$ does not have rank m then there must exist a nonzero m-vector $\bar{v}$ such that $\bar{A}_B^T \bar{v} = 0$. If σ denotes the quantity $\sigma = -a_{\beta_r}^T \bar{v}$, then the $(m+1)$-vector $v = (\bar{v}, \sigma)$ satisfies

$$\begin{pmatrix} a_{\beta_r}^T & 1 \\ \bar{A}_B^T & 0 \end{pmatrix} \begin{pmatrix} \bar{v} \\ \sigma \end{pmatrix} = 0, \quad \text{or equivalently,} \quad \begin{pmatrix} A_B^T & e_r \end{pmatrix} v = 0.$$

The scalar σ must be nonzero or else $A_B^T \bar{v} = 0$, which would contradict the assumption that A_B has rank m. Then

$$v^T \begin{pmatrix} A_B \\ e_r^T \end{pmatrix} p_B = v^T \begin{pmatrix} 0 \\ (p_B)_r \end{pmatrix} = \sigma (p_B)_r = 0,$$

which implies that $(p_B)_r = 0$. This is a contradiction because the ratio test (A.7) will choose β_r as the outgoing basic variable only if $(p_B)_r < 0$. It follows that $\bar{v} = 0$, and hence $\bar{A}_B$ must have rank m.

If H is not positive definite, the KKT matrix K_B associated with the equations (A.11) may have fewer than n_B positive eigenvalues (cf. (A.2)), i.e., the reduced Hessian $Z_B^T H_B Z_B$ may be singular or indefinite. In this situation, the subproblem (A.10) is unbounded and the equations (A.11) cannot be used directly to define p. In this case we seek a direction p such that $p_N = 0$ and $A_B p_B = 0$, where

$$g_B^T p_B < 0, \quad \text{and} \quad p_B^T H_B p_B \le 0. \tag{A.12}$$

The QP objective decreases without bound along such a direction, so either the largest feasible step α_M (A.7) is infinite, or a basic variable must become nonbasic at some finite α_M such that $\widehat{f}(x + \alpha_M p) \le \widehat{f}(x)$. If $\alpha_M = +\infty$, the QP problem is unbounded and the algorithm is terminated.

A number of methods[1] maintain an unsymmetric block-triangular decomposition of K_B in which the reduced Hessian $Z_B^T H_B Z_B$ is one of the diagonal blocks (the precise form of the decomposition is discussed in Section A.4.1). Given this block-triangular decomposition, the methods of Gill and Murray [83], Gill et al. [89,94], and Gill, Murray and Saunders [85] factor the reduced Hessian as $L_B D_B L_B^T$, where L_B is unit lower triangular and D_B is diagonal. These methods control the inertia of K_B by starting the iterations at a subspace minimizer. With this restriction, the reduced Hessian has at most one nonpositive eigenvalue, and the direction p_B is unique up to a scalar multiple. This property allows the computation to be arranged so that D_B has at most one nonpositive element, which always occurs in the last position. The vector p_B is then computed from a triangular system involving the rows and

[1] Some were first proposed for the all-inequality constraint case, but they are easily reformulated for constraints in standard form.

columns of L_B associated with the positive-definite principal submatrix of $Z_B^T H_B Z_B$ (see, e.g., Gill et al. [89, 94] for further details).

The method of Bunch and Kaufman [20] allows the reduced Hessian to have any number of nonpositive eigenvalues in the KKT matrix (and therefore need not be started at a subspace minimizer). In this case, a symmetric indefinite factorization of $Z_B^T H_B Z_B$ is maintained, giving a block diagonal factor D_B with 1×1 or 2×2 diagonal blocks. In the strictly convex case, methods may be defined that employ a *symmetric* block decomposition of K_B, see, e.g., Gill et al. [80].

As the reduced Hessian may not be positive definite, methods that maintain a block-triangular decomposition of K_B must use customized methods to factor and modify $Z_B^T H_B Z_B$ as the iterations proceed. This makes it difficult to apply general-purpose solvers that exploit structure in A and H. Methods that factor the KKT system directly are also problematic because K_B can be singular. Fletcher [61] proposed that any potential singularity be handled by embedding K_B in a larger system that is known to be nonsingular. This idea was extended to sparse KKT equations by Gould [105]. Fletcher and Gould define an inertia-controlling method based on solving a nonsingular bordered system that includes information associated with the variable x_{β_s} that was most recently made basic. The required binding direction p_B may be found by solving the bordered system

$$\begin{pmatrix} H_B & A_B^T & e_s \\ A_B & & \\ e_s^T & & \end{pmatrix} \begin{pmatrix} p_B \\ -q_\pi \\ -\mu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

which is nonsingular. A simple calculation shows that p_B, q_π and μ satisfy

$$\begin{pmatrix} \bar{H}_B & -\bar{A}_B^T \\ \bar{A}_B & 0 \end{pmatrix} \begin{pmatrix} p_B \\ q_\pi \end{pmatrix} = -\begin{pmatrix} (h_{\beta_s})_{\bar{\mathcal{B}}} \\ a_{\beta_s} \end{pmatrix} \quad \text{and} \quad \mu = p_B^T H_B p_B, \tag{A.13}$$

where $\bar{\mathcal{B}}$ is the basic set with index β_s omitted. A comparison of (A.13) with (A.5) shows that their respective values of (p_B, q_π) are the same, which implies that Fletcher-Gould binding direction is identical to the nonbinding direction of Section A.1.1. In fact, all binding and nonbinding direction inertia-controlling methods generate the same sequence of iterates when started at the same subspace minimizer. The only difference is in the order in which the computations are performed—binding-direction methods make the targeted nonbasic variable basic at the start of the sequence of consecutive iterates, whereas nonbinding-direction methods make the variable basic at the end of the sequence when the associated shifted bound constraint has a zero multiplier. However, it must be emphasized that not all QP methods are inertia controlling. Some methods allow any number of zero eigenvalues in the KKT matrix—see, for example, the Bunch-Kaufman method mentioned above, and the QP methods in the GALAHAD software package of Gould, Orban, and Toint [111, 112, 106].

A.2. Dual active-set methods

In the convex case (i.e., when H is positive semidefinite) the dual of the QP subproblem (A.1) is

$$\begin{array}{ll} \underset{w\in\mathbb{R}^n, \pi\in\mathbb{R}^m, z\in\mathbb{R}^n}{\text{minimize}} & \widehat{f}_D(w,\pi,z) = b^T\pi + x_I^T z + \tfrac{1}{2}(w - x_I)^T H (w - x_I) \\ \text{subject to} & H(w - x_I) - A^T\pi - z = -g, \quad z \ge 0. \end{array} \tag{A.14}$$

The dual constraints are in standard form, with nonnegativity constraints on z. The optimality conditions for the dual are characterized by the following result.

Result A.3. (Optimality conditions for the dual)
The point (w, π, z) is a minimizer of the dual QP (A.14) if and only if

(a) (w, π, z) *satisfies* $H(w - x_I) - A^T\pi - z = -g$, *with* $z \geq 0$;

(b) *there exists an n-vector y such that*

  (i) $H(w - x_I) = H(y - x_I)$;

  (ii) $Ay = Ax_I - b$, *with* $y \geq 0$; *and*

  (iii) $y \cdot z = 0$. ▮

The components of y are the Lagrange multipliers for the dual bounds $z \geq 0$. Similarly, the components of $y - x_I$ are the "π-values", i.e., the multipliers for the equality constraints $H(w - x_I) - A^T\pi - z = -g$. The relationship between the primal and dual solution was first given by Dorn [51]. If the dual has a bounded solution, then part (b) implies that the vector y of multipliers for the dual is a KKT point of the primal, and hence constitutes a primal solution. Moreover, if the dual has a bounded solution and H is positive definite, then $w = y$.

A.2.1. A dual nonbinding direction method

Dual active-set methods can be defined that are based on applying conventional primal active-set methods to the dual problem (A.14). For brevity, we consider the case where H is positive definite; the positive semidefinite case is considered by Gill and Wong [98]. Consider a feasible point (w, π, z) for the dual QP (A.14), i.e., $H(w - x_I) - A^T\pi - z = -g$ and $z \geq 0$. Our intention is to make the notation associated with the dual algorithm consistent with the notation for the primal. To do this, we break with the notation of Section A.1 and use $\mathcal{B}$ to denote the *nonbasic* set and $\mathcal{N}$ to denote the *basic* set for the dual QP. This implies that the dual *nonbasic* variables are $\{z_{\beta_1}, z_{\beta_2}, \ldots, z_{\beta_{n_B}}\}$, where $n_B = n - n_N$.

A dual basis contains all the columns of $\begin{pmatrix} H & -A^T \end{pmatrix}$ together with the unit columns corresponding to the dual basic variables, i.e., the columns of I with indices in $\mathcal{N}$. It follows that the rows and columns of the dual basis may be permuted to give

$$\begin{pmatrix} H_B & H_D & -A_B^T & 0 \\ H_D^T & H_N & -A_N^T & -I_N \end{pmatrix}, \tag{A.15}$$

where A_N and A_B denote the columns of A indexed by $\mathcal{N}$ and $\mathcal{B}$. The dual nonbasic set $\mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_{n_B}\}$ now provides an estimate of which of the bounds $z \geq 0$ are active at the solution of (A.14). As H is positive definite, the dual basis has full row rank regardless of the rank of the submatrix $-A_B^T$. This implies that if the columns of A_B are to be used to define a basis for a primal solution, it is necessary to impose additional conditions on the dual basis. Here, we assume that the matrix

$$K_B = \begin{pmatrix} H_B & A_B^T \\ A_B & 0 \end{pmatrix} \tag{A.16}$$

is nonsingular. This condition ensures that A_B has rank m. To distinguish K_B from the full KKT matrix for the dual, we refer to K_B as the *reduced KKT matrix*. The next result concerns the properties of a subspace minimizer for the dual QP.

Result A.4. (Properties of a subspace minimizer for the dual) *Consider the dual QP (A.14) with H positive definite.*

(a) *If (w, π, z) is a subspace stationary point, then there exists a vector x such that*

$$Hw = Hx, \quad \textit{with} \;\; A_B x_B + A_N x_N = Ax_I - b, \quad \textit{and} \;\; x_N = 0.$$

(b) *A dual subspace stationary point at which the reduced KKT matrix (A.16) is nonsingular is a dual subspace minimizer.*

(c) *If (w, π, z) is a standard subspace minimizer, then $z_B = 0$ and $z_N \geq 0$.*

This result implies that $x = w$ at a dual subspace minimizer for the special case of H positive definite. However, it is helpful to distinguish between w and x to emphasize that x is the vector of dual variables for the dual problem. At a subspace stationary point, x is a basic solution of the primal equality constraints. Moreover, $z = H(w - x_I) - A^T\pi + g = \widehat{g}(w) - A^T\pi = \widehat{g}(x) - A^T\pi$, which are the primal reduced-costs.

Let (w, π) be a nonoptimal dual subspace minimizer for the dual QP (A.14). (It will be shown below that the vector w need not be computed explicitly.) As (w, π) is not optimal, there is at least one negative component of the dual multiplier vector x_B, say x_{β_r}. If we apply the nonbinding-direction method of Section A.1.1, we define a dual search direction $(\Delta w, q_\pi, \Delta z)$ that is feasible for the dual equality constraints and increases a nonbasic variable with a negative multiplier. As (w, π, z) is assumed to be dual feasible, this gives the constraints for the equality-constraint QP subproblem in the form

$$H\Delta w - A^T q_\pi - \Delta z = 0, \qquad \Delta z_B = e_r.$$

The equations analogous to (A.4) for the dual direction $(p, q_\pi, \Delta z)$ are

$$\left(\begin{array}{cccccc|c} H_B & H_D & 0 & 0 & -H_B & -H_D & 0 \\ H_D^T & H_N & 0 & 0 & -H_D^T & -H_N & 0 \\ 0 & 0 & 0 & 0 & A_B & A_N & 0 \\ 0 & 0 & 0 & 0 & 0 & I_N & 0 \\ H_B & H_D & -A_B^T & 0 & 0 & 0 & I_B \\ H_D^T & H_N & -A_N^T & -I_N & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & I_B \end{array}\right) \begin{pmatrix} \Delta w_B \\ \Delta w_N \\ q_\pi \\ \Delta z_N \\ p_B \\ p_N \\ \Delta z_B \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ e_r \end{pmatrix},$$

where p_B and p_N denote the changes in the multipliers x of the dual. Block elimination gives $H\Delta w = Hp$, where p_B, p_N and q_π are determined by the equations

$$p_N = 0, \quad \text{and} \quad \begin{pmatrix} H_B & -A_B^T \\ A_B & 0 \end{pmatrix} \begin{pmatrix} p_B \\ q_\pi \end{pmatrix} = \begin{pmatrix} e_r \\ 0 \end{pmatrix}. \tag{A.17}$$

As $H\Delta w = Hp$, the change in z can be computed as $\Delta z = Hp - A^T q_\pi$. The curvature of the dual objective is given by $\Delta w^T H \Delta w = (p_B)_r$ from (A.8). If the curvature is nonzero, the step $\alpha_* = -(x_B)_r/(p_B)_r$ minimizes the dual objective $\widehat{f}_D(w + \alpha\Delta w, \pi + \alpha q_\pi, z + \alpha\Delta z)$ with respect to α, and the rth element of $x_B + \alpha_* p_B$ is zero. If the x_B are interpreted as estimates of the primal variables, the step from x_B to $x_B + \alpha_* p_B$ increases the negative (and hence infeasible) primal variable $(x_B)_r$ until it reaches its bound of zero. If $\alpha = \alpha_*$ gives a feasible point for dual inequalities, i.e., if $z + \alpha_*\Delta z$ are nonnegative, then the new iterate is $(w + \alpha_*\Delta w, \pi + \alpha_* q_\pi, z + \alpha_*\Delta z)$. In this case, the nonbinding variable is removed from the dual nonbasic set, which means that the index β_r is moved to $\mathcal{N}$ and the associated entries of H and A are removed from H_B and A_B.

If $\alpha = \alpha_*$ is unbounded, or $(w + \alpha_*\Delta w, \pi + \alpha_* q_\pi, z + \alpha_*\Delta z)$ is not feasible, the step is the largest α such that $g(w + \alpha\Delta w) - A^T(\pi + \alpha q_\pi)$ is nonnegative. The required value is

$$\alpha_F = \min_{1 \leq i \leq n_N} \{\gamma_i\}, \quad \text{where} \quad \gamma_i = \begin{cases} \dfrac{(z_N)_i}{-(\Delta z_N)_i} & \text{if } (\Delta z_N)_i < 0 \\ +\infty & \text{otherwise.} \end{cases} \tag{A.18}$$

If $\alpha_F < \alpha_*$ then at least one component of z_N is zero at $(w + \alpha_F \Delta w, \pi + \alpha_F q_\pi, z + \alpha_F \Delta z)$, and the index of one of these, ν_s say, is moved to $\mathcal{B}$. The composition of the new dual basis is determined by the singularity vector, adapted to the dual QP from the nonbinding direction method of Section A.1.1. Define the vector u, u_π and v such that $u = v - e_{\nu_s}$, where $v_N = 0$, and u_π and v_B are determined by the equations

$$\begin{pmatrix} H_B & -A_B^T \\ A_B & 0 \end{pmatrix} \begin{pmatrix} v_B \\ u_\pi \end{pmatrix} = \begin{pmatrix} (h_{\nu_s})_B \\ a_{\nu_s} \end{pmatrix}.$$

If u is zero, then $(w + \alpha_F q_\pi, \pi + \alpha_F q_\pi, z + \alpha_F \Delta z)$ is a subspace minimizer with respect to the basis defined with variable β_r replaced by constraint ν_s. Otherwise, ν_s is moved to $\mathcal{B}$, which has the effect of adding the column a_{ν_s} to A_B, and adding a row and column to H_B. As in the primal nonbinding direction method, the vectors p and q_π may be computed in $O(n)$ operations if no column swap is made.

If (w, π, z) is subspace minimizer at which the reduced KKT matrix (A.16) is nonsingular, then the next iterate is also a subspace minimizer with a nonsingular reduced KKT matrix. (For more details, see Gill and Wong [98]).

The algorithm described above is a special case of the method of Gill and Wong [98], which is defined for the general convex case (i.e., when H can be singular). If $H = 0$ this method is equivalent to the dual simplex method. Bartlett and Biegler [6] propose a method for the strictly convex case that uses the Schur-complement method to handle the changes to the KKT equations when the active set changes (see Section A.4.2).

The dual problem (A.14) has fewer inequality constraints than variables, which implies that if H and A have no common nontrivial null vector, then the dual constraint gradients, the rows of $\begin{pmatrix} H & -A^T \end{pmatrix}$, are linearly independent, and the dual feasible region has no degenerate points. In this situation, an active-set dual method cannot cycle, and will either terminate with an optimal solution or declare the dual problem to be unbounded. This nondegeneracy property does not hold for a dual linear program, but it does hold for strictly convex problems and any QP with H and A of the form

$$H = \begin{pmatrix} \bar{H} & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} \bar{A} & -I_m \end{pmatrix},$$

where $\bar{H}$ is an $(n-m) \times (n-m)$ positive-definite matrix.

A.2.2. Finding an initial dual-feasible point

An initial dual-feasible point may be defined by applying a conventional phase-one method to the dual constraints, i.e., by minimizing the sum of infeasibilities for the dual constraints $H(x - x_I) - A^T\pi - z = -g$, $z \geq 0$. If H is nonsingular and A has full rank, another option is to define $\mathcal{N} = \emptyset$ and compute (x_0, π_0) from the equations

$$\begin{pmatrix} H & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ \pi_0 \end{pmatrix} = -\begin{pmatrix} g - Hx_I \\ b - Ax_I \end{pmatrix}.$$

This choice of basis gives $z_0 = 0$, $\widehat{c}(x_0) = 0$, with (x_0, π_0, z_0) a dual subspace minimizer.

A.2.3. The Goldfarb-Idnani method

If H is nonsingular, the vectors

$$y = \begin{pmatrix} \pi \\ z \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} b \\ x_I \end{pmatrix}, \quad \text{and} \quad \bar{A} = \begin{pmatrix} A \\ I \end{pmatrix},$$

may be used to eliminate $w - x_I$ from (A.14) to give the dual problem:

$$\underset{y\in\mathbb{R}^{n+m},\, z\in\mathbb{R}^n}{\text{minimize}} \quad y^T(\bar{b} - \bar{A}H^{-1}g) + \tfrac{1}{2}y^T\bar{A}H^{-1}\bar{A}^Ty, \quad \text{subject to} \quad z \ge 0.$$

Some references include: Goldfarb and Idnani [102], Powell [156]. A variant of the Goldfarb and Idnani method for dense convex QP has been proposed by Boland [11].

A.3. QP regularization

The methods considered above rely on the assumption that each basis matrix A_B has rank m. In an active-set method this condition is guaranteed (at least in exact arithmetic) by the active-set strategy if the *initial* basis has rank m. For methods that solve the KKT system by factoring a subset of m columns of A_B (see Section A.4.1), special techniques can be used to select a linearly independent set of m columns from A. These procedures depend on the method used to factor the basis—for example, the SQP code SNOPT employs a combination of LU factorization and basis repair to determine a full-rank basis. If a factorization reveals that the square submatrix is rank deficient, suspected dependent columns are discarded and replaced by the columns associated with slack variables. However, for methods that solve the KKT system by direct factorization, such as the Schur complement method of Section A.4.2, basis repair is not an option because the factor routine may be a "black-box" that does not incorporate rank-detection. Unfortunately, over the course of many hundreds of iterations, performed with KKT matrices of varying degrees of conditioning, an SQP method can place even the most robust symmetric indefinite solver under considerable stress. (Even a relatively small collection of difficult problems can test the reliability of a solver. Gould, Scott, and Hu [110] report that none of the 9 symmetric indefinite solvers tested was able to solve all of the 61 systems in their collection.) In this situation it is necessary to use a *regularized* method, i.e., a method based on solving equations that are guaranteed to be solvable without the luxury of basis repair.

To illustrate how a problem may be regularized, we start by considering a QP with *equality constraints*, i.e.,

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & g^T(x - x_I) + \tfrac{1}{2}(x - x_I)^TH(x - x_I) \\ \text{subject to} & Ax = Ax_I - b. \end{array} \tag{A.19}$$

Assume for the moment that this subproblem has a feasible primal-dual solution (x^*, π^*). Given an estimate π_E of the QP multipliers π^*, a positive μ and arbitrary ν, consider the *generalized augmented Lagrangian*

$$\begin{aligned}\mathcal{M}(x,\pi;\pi_E,\mu,\nu) = \widehat{f}(x) - \widehat{c}(x)^T\pi_E + \frac{1}{2\mu}\|\widehat{c}(x)\|_2^2 \\ + \frac{\nu}{2\mu}\|\widehat{c}(x) + \mu(\pi - \pi_E)\|_2^2\end{aligned} \tag{A.20}$$

(see Forsgren and Gill [72], and Gill and Robinson [97]). The function $\mathcal{M}$ involves $n + m$ variables and has gradient vector

$$\nabla\mathcal{M}(x,\pi;\pi_E,\mu,\nu) = \begin{pmatrix} \widehat{g}(x) - A^T\pi + (1+\nu)A^T\big(\pi - \bar{\pi}(x)\big) \\ \nu\mu\big(\pi - \bar{\pi}(x)\big) \end{pmatrix}, \tag{A.21}$$

where $\bar{\pi}(x) = \pi_E - \widehat{c}(x)/\mu$. If we happen to know the value of π^*, and define $\pi_E = \pi^*$, then simple substitution in (A.21) shows that (x^*, π^*) is a stationary point of $\mathcal{M}$ for all ν and all positive μ. The Hessian of $\mathcal{M}$ is given by

$$\nabla^2\mathcal{M}(x,\pi;\pi_E,\mu,\nu) = \begin{pmatrix} H + \big(\frac{1+\nu}{\mu}\big)A^TA & \nu A^T \\ \nu A & \nu\mu I \end{pmatrix}, \tag{A.22}$$

which is independent of π_E. If we make the additional assumptions that ν is *nonnegative* and the reduced Hessian of the QP subproblem is positive definite, then $\nabla^2\mathcal{M}$ is positive semidefinite for all μ sufficiently small. Under these assumptions, if $\pi_E = \pi^*$ it follows that (x^*, π^*) is the unique minimizer of the unconstrained problem

$$\underset{x\in\mathbb{R}^n,\pi\in\mathbb{R}^m}{\text{minimize}} \;\; \mathcal{M}(x, \pi; \pi_E, \mu, \nu) \tag{A.23}$$

(see, e.g., Gill and Robinson [97], Gill and Wong [99]). This result implies that if π_E is an approximate multiplier vector (e.g., from the previous QP subproblem in the SQP context), then the minimizer of $\mathcal{M}(x, \pi; \pi_E, \mu, \nu)$ will approximate the minimizer of (A.19). In order to distinguish between a solution of (A.19) and a minimizer of (A.23) for an arbitrary π_E, we use (x_*, π_*) to denote a minimizer of $\mathcal{M}(x, \pi; \pi_E, \mu, \nu)$. Observe that stationarity of $\nabla\mathcal{M}$ at (x_*, π_*) implies that $\pi_* = \bar\pi(x_*) = \pi_E - \widehat{c}(x_*)/\mu$. The components of $\bar\pi(x_*)$ are the so-called *first-order multipliers* associated with a minimizer of (A.23).

Particular values of the parameter ν give some well-known functions (although, as noted above, each function defines a problem with the common solution (x_*, π_*)). If $\nu = 0$, then $\mathcal{M}$ is independent of π, with

$$\mathcal{M}(x; \pi_E, \mu) \equiv \mathcal{M}(x; \pi_E, \mu, 0) = \widehat{f}(x) - \widehat{c}(x)^T\pi_E + \frac{1}{2\mu}\|\widehat{c}(x)\|_2^2. \tag{A.24}$$

This is the conventional Hestenes-Powell augmented Lagrangian (1.11) applied to (A.19). If $\nu = 1$ in (A.20), $\mathcal{M}$ is the primal-dual augmented Lagrangian

$$\widehat{f}(x) - \widehat{c}(x)^T\pi_E + \frac{1}{2\mu}\|\widehat{c}(x)\|_2^2 + \frac{1}{2\mu}\|\widehat{c}(x) + \mu(\pi - \pi_E)\|_2^2 \tag{A.25}$$

considered by Robinson [157] and Gill and Robinson [97]. If $\nu = -1$, then $\mathcal{M}$ is the proximal-point Lagrangian

$$\widehat{f}(x) - \widehat{c}(x)^T\pi - \frac{\mu}{2}\|\pi - \pi_E\|_2^2.$$

As ν is negative in this case, $\nabla^2\mathcal{M}$ is indefinite and $\mathcal{M}$ has an unbounded minimizer. Nevertheless, a unique minimizer of $\mathcal{M}$ for $\nu > 0$ is a saddle-point for an $\mathcal{M}$ defined with a negative ν. Moreover, for $\nu = -1$, (x^*, π^*) solves the min-max problem

$$\min_x \max_\pi \;\; \widehat{f}(x) - \widehat{c}(x)^T\pi - \frac{\mu}{2}\|\pi - \pi_E\|_2^2.$$

In what follows, we use $\mathcal{M}(v)$ to denote $\mathcal{M}$ as a function of the primal-dual variables $v = (x, \pi)$ for given values of π_E, μ and ν. Given the initial point $v_I = (x_I, \pi_I)$, the stationary point of $\mathcal{M}(v)$ is $v_* = v_I + \varDelta v$, where $\varDelta v = (p, q)$ with $\nabla^2\mathcal{M}(v_I)\varDelta v = -\nabla\mathcal{M}(v_I)$. It can be shown that $\varDelta v$ satisfies the equivalent system

$$\begin{pmatrix} H & -A^T \\ A & \mu I \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} \widehat{g}(x_I) - A^T\pi_I \\ \widehat{c}(x_I) + \mu(\pi_I - \pi_E) \end{pmatrix}, \tag{A.26}$$

which is independent of the value of ν (see Gill and Robinson [97]). If $\nu \neq 0$, the primal-dual direction is unique. If $\nu = 0$ (i.e., $\mathcal{M}$ is the conventional augmented Lagrangian (A.24)), $\varDelta v$ satisfies the equations

$$\begin{pmatrix} H & -A^T \\ A & \mu I \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} \widehat{g}(x_I) - A^T\pi \\ \widehat{c}(x_I) + \mu(\pi - \pi_E) \end{pmatrix}, \tag{A.27}$$

for an *arbitrary* vector π. In this case, p is unique but q depends on the choice of π. In particular, if we define the equations (A.27) with $\pi = \pi_I$, then we obtain directions identical to those of (A.26). Clearly, it must hold that *p is independent of the choice of ν* in (A.20).

The point $(x_*, \pi_*) = (x_I + p, \pi_I + q)$ is the primal-dual solution of the *perturbed QP*

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & g^T(x - x_I) + \tfrac{1}{2}(x - x_I)^T H(x - x_I) \\ \text{subject to} & Ax = Ax_I - b - \mu(\pi_* - \pi_E), \end{array} \tag{A.28}$$

where the perturbation *shifts* each constraint of (A.19) by an amount that depends on the corresponding component of $\pi_* - \pi_E$. Observe that the constraint shift depends on the solution, so it cannot be defined *a priori*. The effect of the shift is to *regularize* the KKT equations by introducing the *nonzero* $(2, 2)$ block μI. In the regularized case *it is not necessary for A to have full row rank for the KKT equations to be nonsingular.* A full-rank assumption is required if the $(2, 2)$ block is zero. In particular, if we choose $\pi_E = \pi_I$, the system (A.26) is:

$$\begin{pmatrix} H & -A^T \\ A & \mu I \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} \widehat{g}(x_I) - A^T \pi_I \\ \widehat{c}(x_I) \end{pmatrix}. \tag{A.29}$$

These equations define a regularized version of the Newton equations (2.7). These equations also form the basis for the primal-dual formulations of the quadratic penalty method considered by Gould [104] (for related methods, see Murray [143], Biggs [8] and Tapia [171]).

The price paid for the regularized equations is an *approximate* solution of the original problem. However, once (x_*, π_*) has been found, π_E can be redefined as π_* and the process repeated—with a smaller value of μ if necessary. There is more discussion of the choice of π_E below. However, before turning to the inequality constraint case, we summarize the regularization for equality constraints.

- The primal-dual solution (x^*, π^*) of the equality constraint problem (A.19) is approximated by the solution of the perturbed KKT system (A.26).

- The resulting approximation $(x_*, \pi_*) = (x_I + p, \pi_I + q)$ is a stationary point of the function $\mathcal{M}$ (A.20) for all values of the parameter ν. If $\mu > 0$ and $\nu \geq 0$ then (x_*, π_*) is a minimizer of $\mathcal{M}$ for all μ sufficiently small.

As the solution of the regularized problem is independent of ν, there is little reason to use nonzero values of ν in the equality-constraint case. However, the picture changes when there are *inequality constraints* and an *approximate* solution of the QP problem is required, as is often the case in the SQP context.

The method defined above can be extended to the inequality constraint problem (A.1) by solving, the bound-constrained subproblem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; \mathcal{M}(x; \pi_E, \mu) \quad \text{subject to} \;\; x \geq 0. \tag{A.30}$$

This technique has been proposed for general nonlinear programming (see, e.g., Conn, Gould and Toint [40,41,42], Friedlander [74], and Friedlander and Saunders [76]), and to quadratic programming (see, e.g., Dostál, Friedlander and Santos [52,53,54], Delbos and Gilbert [47], Friedlander and Leyffer [75]), and Maes [133]). Subproblem (A.30) may be solved using one of the active-set methods of Sections A.1.2 and A.1.1, although no explicit phase-one procedure is needed because there are no general constraints. In the special case of problem (A.30) a primal active-set method defines a sequence of nonnegative iterates $\{x_j\}$ such that $x_{j+1} = x_j + \alpha_j p_j \geq 0$. At the jth iteration of the binding-direction method of Section A.1.2, variables in the nonbasic set $\mathcal{N}$ remain unchanged for any value of the step length, i.e.,

$p_N = (p_j)_N = 0$. This implies that the elements p_B of the direction p_j must solve the unconstrained QP subproblem:

$$\underset{p_B}{\text{minimize}}\;\; p_B^T(\nabla\mathcal{M}_j)_B + \tfrac{1}{2}p_B^T(\nabla^2\mathcal{M}_j)_B p_B.$$

As in (A.26), the optimality conditions imply that p_B satisfies

$$\begin{pmatrix} H_B & -A_B^T \\ A_B & \mu I \end{pmatrix}\begin{pmatrix} p_B \\ q_j \end{pmatrix} = -\begin{pmatrix} (\widehat{g}(x_j) - A^T\pi_j)_B \\ \widehat{c}(x_j) + \mu(\pi_j - \pi_E) \end{pmatrix}, \tag{A.31}$$

where π_j is an estimate of the optimal multipliers π_* of (A.30). The next iterate is defined as $x_{j+1} = x_j + \alpha_j p_j$, where the steplength α_j is chosen such that $x_j + \alpha_j p_j \geq 0$. As in the equality-constraint case, the dual variables may be updated as $\pi_{j+1} = \pi_j + \alpha_j q_j$. The dual iterates π_j will converge to the multipliers π_* of the perturbed QP:

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & g^T(x - x_I) + \tfrac{1}{2}(x - x_I)^T H(x - x_I) \\ \text{subject to} & Ax = Ax_I - b - \mu(\pi_* - \pi_E), \quad x \geq 0. \end{array} \tag{A.32}$$

At an optimal solution (x_*, π_*) of (A.30) the vector $z_* = \widehat{g}(x_*) - A^T\pi_*$ provides an estimate of the optimal reduced costs z^*. As in the equality-constraint case, the vector of first-order multipliers $\bar{\pi}(x_*) = \pi_E - \widehat{c}(x_*)/\mu$ is identical to π_*. Problem (A.30) may be solved using a bound-constraint variant of the nonbinding-direction method of Section A.1.1. This method has some advantages when solving convex and general QP problems. For more details, see Gill and Wong [99].

If the QP is a “one-off” problem, then established techniques associated with the bound-constrained augmented Lagrangian method can be used to update π_E and μ (see, e.g., Conn, Gould and Toint [41], Dostál, Friedlander and Santos [52, 53, 54], Delbos and Gilbert [47], Friedlander and Leyffer [75], Maes [133], and Gill and Wong [98]). These rules are designed to update π_E and μ without the need to find the exact solution of (A.30). In the SQP context, it may be more appropriate to find an approximate solution of (A.30) for a *fixed* value of π_E, which is then updated in the outer iteration. Moreover, as μ is being used principally for regularization, it is given a smaller value than is typical in a conventional augmented Lagrangian method.

The algorithms defined above are *dual* regularization methods in the sense that the regularization has the effect of bounding the Lagrange multipliers. For convex QP certain *primal* regularization schemes may be used to bound the primal variables (see, e.g., Gill et al. [84], Saunders [161], Saunders and Tomlin [163, 162], Altman and Gondzio [1], and Maes [133]).

A.4. Solving the KKT system

The principal work associated with a QP iteration is the cost of solving one or two saddle-point systems of the form

$$\begin{pmatrix} H_B & -A_B^T & H_D \\ A_B & \mu I & A_N \\ & & I_N \end{pmatrix}\begin{pmatrix} y_B \\ w \\ y_N \end{pmatrix} = \begin{pmatrix} g_B \\ f_1 \\ f_2 \end{pmatrix}, \tag{A.33}$$

where μ is a nonnegative scalar. We focus on two approaches appropriate for large-scale quadratic programming.

A.4.1. Variable-reduction methods

These methods are appropriate for the case $\mu = 0$. As A_B has rank m, there exists a nonsingular Q_B such that

$$A_B Q_B = \begin{pmatrix} 0 & B \end{pmatrix}, \tag{A.34}$$

with B an $m \times m$ nonsingular matrix. If $\mu = 0$ the matrix Q_B is used to transform the generic system (A.33) to block-triangular form. The columns of Q_B are partitioned so that $Q_B = \begin{pmatrix} Z_B & Y_B \end{pmatrix}$ with Z_B an $n_B - m$ by n_B matrix, then $A_B Z_B = 0$ and the columns of Z_B span the null-space of A_B. Analogous to (2.11) we obtain the permuted block-triangular system:

$$\begin{pmatrix} Z_B^T H_B Z_B & Z_B^T H_B Y_B & & Z_B^T H_D \\ Y_B^T H_B Z_B & Y_B^T H_B Y_B & -B^T & Y_B^T H_D \\ & B & 0 & A_N \\ & & & I_N \end{pmatrix} \begin{pmatrix} y_Z \\ y_Y \\ w \\ y_N \end{pmatrix} = \begin{pmatrix} g_Z \\ g_Y \\ f_1 \\ f_2 \end{pmatrix}, \tag{A.35}$$

with $g_Z = Z_B^T g_B$ and $g_Y = Y_B^T g_B$. We formulate the result of the block substitution in a form that uses matrix-vector products involving the full matrix H rather than the submatrix H_B. This is done to emphasize the practical utility of accessing the QP Hessian as an *operator* that defines the product Hx for a given x. This reformulation requires the definition of the explicit column permutation P that identifies the basic and nonbasic columns A_B and A_N of A, i.e.,

$$AP = \begin{pmatrix} A_B & A_N \end{pmatrix}. \tag{A.36}$$

Given the permutation P, we define matrices Q, W^T, Y^T and Z^T that act on vectors of length n, i.e., $Q = \begin{pmatrix} Z & Y & W \end{pmatrix}$, where

$$Z = P\begin{pmatrix} Z_B \\ 0 \end{pmatrix}, \quad Y = P\begin{pmatrix} Y_B \\ 0 \end{pmatrix}, \quad \text{and} \quad W = P\begin{pmatrix} 0 \\ I_N \end{pmatrix}.$$

In terms of these matrices, the block substitution yields

$$\begin{aligned} y_W &= f_2, & y_0 &= W y_W, \\ B y_Y &= f_1 - A_N f_2, & y_1 &= Y y_Y + y_0, \\ Z^T H Z y_Z &= Z^T(g - H y_1), & y &= Z y_Z + y_1, \\ B^T w &= -Y^T(g - Hy). \end{aligned} \tag{A.37}$$

There are many practical choices for the matrix Q_B. For small-to-medium scale problems with dense A and H, the matrix Q_B can be calculated as the orthogonal factor associated with the QR factorization of a row and column-permuted A_B^T (see, e.g., Gill et al. [89]). The method of *variable reduction* is appropriate when A is sparse. In this case the permutation P of (A.36) is specialized further to give

$$AP = \begin{pmatrix} A_B & A_N \end{pmatrix}, \quad \text{with} \quad A_B = \begin{pmatrix} B & S \end{pmatrix},$$

where B an $m \times m$ nonsingular subset of the columns of A and S an $m \times n_S$ matrix with $n_S = n_B - m$. The matrix $Q = \begin{pmatrix} Z & Y & W \end{pmatrix}$ is constructed so that

$$Z = P\begin{pmatrix} -B^{-1}S \\ I_{n_S} \\ 0 \end{pmatrix}, \quad Y = P\begin{pmatrix} I_m \\ 0 \\ 0 \end{pmatrix}, \quad \text{and} \quad W = P\begin{pmatrix} 0 \\ 0 \\ I_N \end{pmatrix}.$$

This form means that matrix-vector products $Z^T v$ or Zv can be computed using a factorization of B (typically, a sparse LU factorization; see Gill et al. [91]), and Z need not be stored explicitly.

A.4.2. The Schur complement method

Solving a "one-off" KKT system can be done very effectively using sparse matrix factorization techniques. However, within a QP algorithm, many systems must be solved in which the matrix changes by a single row and column. Instead of redefining and re-solving the KKT equations at each iteration, the solution may be found by solving a bordered system of the form

$$\begin{pmatrix} K_0 & V \\ V^T & D \end{pmatrix} \begin{pmatrix} z \\ w \end{pmatrix} = \begin{pmatrix} b \\ f \end{pmatrix}, \quad \text{with } K_0 = \begin{pmatrix} H_{\scriptscriptstyle B} & A_{\scriptscriptstyle B}^T \\ A_{\scriptscriptstyle B} & \mu I \end{pmatrix}, \tag{A.38}$$

where K_0 is the KKT matrix at the initial point. For simplicity, we assume that the second block of the variables is scaled by -1 so that the $(1,2)$ block of K_0 is $A_{\scriptscriptstyle B}^T$, not $-A_{\scriptscriptstyle B}^T$. The *Schur complement method* is based on the assumption that factorizations for K_0 and the *Schur complement* $C = D - V^T K_0^{-1} V$ exist. Then the solution of (A.38) can be determined by solving the equations

$$K_0 t = b, \qquad Cw = f - V^T t, \qquad K_0 z = b - Vw.$$

The work required is dominated by two solves with the fixed matrix K_0 and one solve with the Schur complement C. If the number of changes to the basic set is small enough, dense factors of C may be maintained.

We illustrate the definition of (A.38) immediately after the matrix K_0 is factorized. (For more details, see, e.g., Bisschop and Meeraus [9], Gill et al. [93].) Suppose that variable s enters the basic set. The next KKT matrix can be written as

$$\left(\begin{array}{cc|c} H_{\scriptscriptstyle B} & A_{\scriptscriptstyle B}^T & (h_s)_{\scriptscriptstyle B} \\ A_{\scriptscriptstyle B} & \mu I & a_s \\ \hline (h_s)_{\scriptscriptstyle B}^T & a_s^T & h_{ss} \end{array}\right),$$

where a_s and h_s are the sth columns of A and H. This is a matrix of the form (A.38) with $D = (h_{ss})$ and $V^T = \big((h_s)_{\scriptscriptstyle B}^T \quad a_s^T\big)$.

Now consider the case where the rth basic variable is deleted from the basic set, so that the rth column is removed from $A_{\scriptscriptstyle B}$. The corresponding changes can be enforced in the solution of the KKT system using the bordered matrix:

$$\left(\begin{array}{ccc|c} H_{\scriptscriptstyle B} & A_{\scriptscriptstyle B}^T & (h_s)_{\scriptscriptstyle B} & e_r \\ A_{\scriptscriptstyle B} & \mu I & a_s & 0 \\ (h_s)_{\scriptscriptstyle B}^T & a_s^T & h_{ss} & 0 \\ \hline e_r^T & 0 & 0 & 0 \end{array}\right).$$

Bordering with the unit row and column has the effect of zeroing out the components of the solution corresponding to the deleted basic variable.

The Schur complement method can be extended to a *block LU method* by storing the bordered matrix in block-factored form

$$\begin{pmatrix} K_0 & V \\ V^T & D \end{pmatrix} = \begin{pmatrix} L & \\ Z^T & I \end{pmatrix} \begin{pmatrix} U & Y \\ & C \end{pmatrix}, \tag{A.39}$$

where $K_0 = LU$, $LY = V$, $U^T Z = V$, and $C = D - Z^T Y$, which is the Schur complement matrix (see Eldersveld and Saunders [57], Huynh [125]).

Using the block factors, the solution of (A.38) can be computed from the equations

$$Lt = b, \qquad Cw = f - Z^T t, \qquad Uz = t - Yw.$$

This method requires a solve with L and U each, one multiply with Y and Z^T, and one solve with the Schur complement C.

Although the augmented system (in general) increases in dimension by one at every iteration, the K_0 block is fixed and defined by the initial basic set. As the inner iterations proceed, the size of C increases and the work required to perform a solve or update increases. It may be necessary to restart the process by discarding the existing factors and re-forming K_0 based on the current set of basic variables.

A.5. Finding an initial feasible point

There are two approaches to finding a feasible point for the QP constraints. The first, common in linear programming, is to find a point that satisfies the equality constraints and then iterate (if necessary) to satisfy the bounds. The second method finds a point that satisfies the bound constraints and then iterates to satisfy the equality constraints. In each case we assume that an initial nonbasic set is known (in the SQP context, this is often the final nonbasic set from the previous QP subproblem).

The first approach is suitable if the variable reduction method is used in the optimality phase. In this case, a factorization is available of the matrix B such that $A_B Q_B = \begin{pmatrix} 0 & B \end{pmatrix}$. Given x_I, a point x_0 is computed that satisfies the constraints $Ax = Ax_I - b$, i.e., we define

$$Bp_Y = -\widehat{c}(x_I), \quad p_F = Yp_Y, \quad x_0 = x_I + p_F.$$

If $x_0 \not\geq 0$, then x_0 is the initial iterate for a phase-one algorithm that minimizes the linear function $-\sum_{i\in\mathcal{V}(x)} x_i$, where $\mathcal{V}(x)$ is the index set of violated bounds at x. The two-phase nature of the algorithm is reflected by changing the function being minimized from the sum of infeasibilities to the quadratic objective function. The function $-\sum_{i\in\mathcal{V}(x)} x_i$ is a piece-wise linear function that gives the one-norm of the constraint infeasibilities at x. A feature of this approach is that many violated constraints can become feasible at any given iteration.

Minimizing the explicit sum of infeasibilities directly is less straightforward for the Schur complement method because the objective function changes from one iteration to the next. In this case, a single phase with a composite objective may be used, i.e.,

$$\begin{array}{ll} \underset{x,v,w}{\text{minimize}} & g^T(x - x_I) + \tfrac{1}{2}(x - x_I)^T H(x - x_I) + e^T u + e^T v \\ \text{subject to} & Ax - u + v = Ax_I - b, \quad x \geq 0, \quad u \geq 0, \quad v \geq 0, \end{array} \tag{A.40}$$

where e is the vector of ones. This approach has been used by Gould [105], and Huynh [125]. An alternative is to define a phase-one subproblem that minimizes the two-norm of the constraint violations, i.e.,

$$\underset{x,v}{\text{minimize}} \;\; \tfrac{1}{2}\|v\|_2^2 \quad \text{subject to} \quad Ax + v = Ax_I - b, \quad x \geq 0. \tag{A.41}$$

This problem is a convex QP. Given an initial point x_0 and nonbasic set $\mathcal{N}_0$ for the phase-two problem, the basic variables for phase one consist of the x_0 variables in $\mathcal{B}_0$ and the m variables v_0 such that $Ax_0 + v_0 = Ax_I - b$. The variables v are always basic.

Another phase-one subproblem appropriate for the Schur complement method minimizes a strictly convex objective involving the two-norm of the constraint violations and a primal regularization term:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; \tfrac{1}{2}\|Ax - b\|_2^2 + \tfrac{1}{2}\sigma\|x - x_I\|_2^2 \quad \text{subject to} \quad x \geq 0. \tag{A.42}$$

If a feasible point exists, then the problem is feasible with the objective bounded below by zero. The only constraints of this problem are bounds on x. Applying the nonbinding direction method of Section A.1.1 gives

$$\begin{pmatrix} \sigma I_B & A_B^T \\ A_B & -I \end{pmatrix} \begin{pmatrix} p_B \\ -q_\pi \end{pmatrix} = \begin{pmatrix} 0 \\ -a_{\nu_s} \end{pmatrix},$$

with $p_N = e_s$ and $q_N = \sigma e_s - A_N^T q_\pi$. Solving this phase-one problem is equivalent to applying the regularized QP algorithm in [99] with $\mu = 1$ and $\pi_E = 0$, to the problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; \frac{\sigma}{2}\|x - x_0\|_2^2 \;\; \text{subject to} \;\; Ax = b, \;\; x \geq 0.$$

References

[1] A. Altman and J. Gondzio. Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. *Optim. Methods Softw.*, 11/12(1-4):275–302, 1999.

[2] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41 (electronic), 2001.

[3] M. Anitescu. On the rate of convergence of sequential quadratic programming with nondifferentiable exact penalty function in the presence of constraint degeneracy. *Math. Program.*, 92(2, Ser. A):359–386, 2002.

[4] M. Anitescu. A superlinearly convergent sequential quadratically constrained quadratic programming algorithm for degenerate nonlinear programming. *SIAM J. Optim.*, 12(4):949–978, 2002.

[5] C. Ashcraft and R. Grimes. SPOOLES: an object-oriented sparse matrix library. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing 1999 (San Antonio, TX)*, page 10, Philadelphia, PA, 1999. SIAM.

[6] R. A. Bartlett and L. T. Biegler. QPSchur: a dual, active-set, Schur-complement method for large-scale and structured convex quadratic programming. *Optim. Eng.*, 7(1):5–32, 2006.

[7] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Athena Scientific, Belmont, Massachusetts, 1996.

[8] M. C. Biggs. Constrained minimization using recursive equality quadratic programming. In F. A. Lootsma, editor, *Numerical Methods for Nonlinear Optimization*, pages 411–428. Academic Press, London and New York, 1972.

[9] J. Bisschop and A. Meeraus. Matrix augmentation and partitioning in the updating of the basis inverse. *Math. Program.*, 13:241–254, 1977.

[10] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. In *Acta Numerica, 1995*, volume 4 of *Acta Numer.*, pages 1–51. Cambridge Univ. Press, Cambridge, 1995.

[11] N. L. Boland. A dual-active-set algorithm for positive semi-definite quadratic programming. *Math. Programming*, 78(1, Ser. A):1–27, 1997.

[12] J. M. Borwein. Necessary and sufficient conditions for quadratic minimality. *Numer. Funct. Anal. and Optimiz.*, 5:127–140, 1982.

[13] A. M. Bradley. *Algorithms for the Equilibration of Matrices and Their Application to Limited-Memory Quasi-Newton Methods*. PhD thesis, Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, May 2010.

[14] K. W. Brodlie, A. R. Gourlay, and J. Greenstadt. Rank-one and rank-two corrections to positive definite matrices expressed in product form. *J. Inst. Math. Appl.*, 11:73–82, 1973.

[15] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. Parts I and II. *J. Inst. Maths. Applns.*, 6:76–90 and 222–231, 1970.

[16] A. Buckley and A. LeNir. QN-like variable storage conjugate gradients. *Math. Program.*, 27:155–175, 1983.

[17] A. Buckley and A. LeNir. BBVSCG–a variable storage algorithm for function minimization. *ACM Trans. Math. Software*, 11:103–119, 1985.

[18] J. R. Bunch. Partial pivoting strategies for symmetric matrices. *SIAM J. Numer. Anal.*, 11:521–528, 1974.

[19] J. R. Bunch and L. Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Math. Comput.*, 31:163–179, 1977.

[20] J. R. Bunch and L. Kaufman. A computational method for the indefinite quadratic programming problem. *Linear Algebra Appl.*, 34:341–370, 1980.

[21] J. R. Bunch and B. N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 8:639–655, 1971.

[22] J. V. Burke. A sequential quadratic programming method for potentially infeasible mathematical programs. *J. Math. Anal. Appl.*, 139(2):319–351, 1989.

[23] J. V. Burke. A robust trust region method for constrained nonlinear programming problems. *SIAM J. Optim.*, 2(2):324–347, 1992.

[24] J. V. Burke and S.-P. Han. A robust sequential quadratic programming method. *Math. Programming*, 43(3, (Ser. A)):277–303, 1989.

[25] R. H. Byrd, F. E. Curtis, and J. Nocedal. Infeasibility detection and SQP methods for nonlinear optimization. *SIAM Journal on Optimization*, 20(5):2281–2299, 2010.

[26] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. *Math. Program.*, 100(1, Ser. B):27–48, 2004.

[27] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. On the convergence of successive linear-quadratic programming algorithms. *SIAM J. Optim.*, 16(2):471–489, 2005.

[28] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-Newton matrices and their use in limited-memory methods. *Math. Program.*, 63:129–156, 1994.

[29] R. H. Byrd, J. Nocedal, and R. A. Waltz. Steering exact penalty methods for nonlinear programming. *Optim. Methods Softw.*, 23(2):197–213, 2008.

[30] R. H. Byrd, R. A. Tapia, and Y. Zhang. An SQP augmented Lagrangian BFGS algorithm for constrained optimization. *SIAM J. Optim.*, 20:210–241, 1992.

[31] R. M. Chamberlain, M. J. D. Powell, C. Lemarechal, and H. C. Pedersen. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Math. Programming Stud.*, (16):1–17, 1982. Algorithms for constrained minimization of smooth nonlinear functions.

[32] C. M. Chin. *A New Trust Region based SLP Filter Algorithm which uses EQP Active Set Strategy.* PhD thesis, Department of Mathematics, University of Dundee, Scotland, 2001.

[33] C. M. Chin and R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Math. Program.*, 96(1, Ser. A):161–177, 2003.

[34] C. M. Chin, A. H. A. Rashid, and K. M. Nor. A combined filter line search and trust region method for nonlinear programming. *WSEAS Trans. Math.*, 5(6):656–662, 2006.

[35] J. W. Chinneck. Analyzing infeasible nonlinear programs. *Comput. Optim. Appl.*, 4(2):167–179, 1995.

[36] J. W. Chinneck. *Feasibility and infeasibility in optimization: algorithms and computational methods.* International Series in Operations Research & Management Science, 118. Springer, New York, 2008.

[37] T. F. Coleman and A. R. Conn. On the local convergence of a quasi-Newton method for the nonlinear programming problem. *SIAM J. Numer. Anal.*, 21:775–769, 1984.

[38] T. F. Coleman and A. Pothen. The null space problem I. Complexity. *SIAM J. on Algebraic and Discrete Methods*, 7:527–537, 1986.

[39] T. F. Coleman and D. C. Sorensen. A note on the computation of an orthogonal basis for the null space of a matrix. *Math. Program.*, 29:234–242, 1984.

[40] A. R. Conn, N. I. M. Gould, and P. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM J. Numer. Anal.*, 25:433–460, 1988.

[41] A. R. Conn, N. I. M. Gould, and P. L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Numer. Anal.*, 28:545–572, 1991.

[42] A. R. Conn, N. I. M. Gould, and P. L. Toint. `LANCELOT`*: a Fortran package for large-scale nonlinear optimization (Release A).* Lecture Notes in Computation Mathematics 17. Springer Verlag, Berlin, Heidelberg, New York, London, Paris and Tokyo, 1992.

[43] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[44] L. B. Contesse. Une caractérisation complète des minima locaux en programmation quadratique. *Numer. Math.*, 34:315–332, 1980.

[45] R. W. Cottle, G. J. Habetler, and C. E. Lemke. On classes of copositive matrices. *Linear Algebra Appl.*, 3:295–310, 1970.

[46] Y.-H. Dai and K. Schittkowski. A sequential quadratic programming algorithm with non-monotone line search. *Pac. J. Optim.*, 4(2):335–351, 2008.

[47] F. Delbos and J. C. Gilbert. Global linear convergence of an augmented Lagrangian algorithm to solve convex quadratic optimization problems. *J. Convex Anal.*, 12(1):45–69, 2005.

[48] R. S. Dembo and U. Tulowitzki. Sequential truncated quadratic programming methods. In *Numerical optimization, 1984 (Boulder, Colo., 1984)*, pages 83–101. SIAM, Philadelphia, PA, 1985.

[49] J. E. Dennis, Jr. and R. B. Schnabel. A new derivation of symmetric positive definite secant updates. In *Nonlinear Programming, 4 (Proc. Sympos., Special Interest Group on Math. Programming, Univ. Wisconsin, Madison, Wis., 1980)*, pages 167–199. Academic Press, New York, 1981.

[50] G. DiPillo and L. Grippo. A new class of augmented Lagrangians in nonlinear programming. *SIAM J. Control Optim.*, 17:618–628, 1979.

[51] W. S. Dorn. Duality in quadratic programming. *Quart. Appl. Math.*, 18:155–162, 1960/1961.

[52] Z. Dostál, A. Friedlander, and S. A. Santos. Adaptive precision control in quadratic programming with simple bounds and/or equalities. In *High performance algorithms and software in nonlinear optimization (Ischia, 1997)*, volume 24 of *Appl. Optim.*, pages 161–173. Kluwer Acad. Publ., Dordrecht, 1998.

[53] Z. Dostál, A. Friedlander, and S. A. Santos. Augmented Lagrangians with adaptive precision control for quadratic programming with equality constraints. *Comput. Optim. Appl.*, 14(1):37–53, 1999.

[54] Z. Dostál, A. Friedlander, and S. A. Santos. Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints. *SIAM J. Optim.*, 13(4):1120–1140 (electronic), 2003.

[55] I. S. Duff. MA57—a code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans. Math. Software*, 30(2):118–144, 2004.

[56] I. S. Duff and J. K. Reid. MA27: a set of Fortran subroutines for solving sparse symmetric sets of linear equations. Technical Report R-10533, Computer Science and Systems Division, AERE Harwell, Oxford, England, 1982.

[57] S. K. Eldersveld and M. A. Saunders. A block-LU update for large-scale linear programming. *SIAM J. Matrix Anal. Appl.*, 13:191–201, 1992.

[58] O. Exler and K. Schittkowski. A trust region SQP algorithm for mixed-integer nonlinear programming. *Optim. Lett.*, 1(3):269–280, 2007.

[59] A. Fischer. Modified Wilson's method for nonlinear programs with nonunique multipliers. *Math. Oper. Res.*, 24(3):699–727, 1999.

[60] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

[61] R. Fletcher. A general quadratic programming algorithm. *J. Inst. Math. Applics.*, 7:76–91, 1971.

[62] R. Fletcher. A model algorithm for composite nondifferentiable optimization problems. *Math. Programming Stud.*, (17):67–76, 1982. Nondifferential and variational techniques in optimization (Lexington, Ky., 1980).

[63] R. Fletcher. Second order corrections for nondifferentiable optimization. In *Numerical analysis (Dundee, 1981)*, volume 912 of *Lecture Notes in Math.*, pages 85–114. Springer, Berlin, 1982.

[64] R. Fletcher. An ℓ_1 penalty method for nonlinear constraints. In P. T. Boggs, R. H. Byrd, and R. B. Schnabel, editors, *Numerical Optimization 1984*, pages 26–40, Philadelphia, 1985.

[65] R. Fletcher. *Practical methods of optimization.* Wiley-Interscience [John Wiley & Sons], New York, 2001.

[66] R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint, and A. Wächter. Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming. *SIAM J. Optim.*, 13(3):635–659 (electronic) (2003), 2002.

[67] R. Fletcher and S. Leyffer. User manual for filterSQP. Technical Report NA/181, Dept. of Mathematics, University of Dundee, Scotland, 1998.

[68] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Math. Program.*, 91(2, Ser. A):239–269, 2002.

[69] R. Fletcher, S. Leyffer, and P. L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optim.*, 13(1):44–59 (electronic), 2002.

[70] R. Fletcher and E. Sainz de la Maza. Nonlinear programming and nonsmooth optimization by successive linear programming. *Math. Program.*, 43:235–256, 1989.

[71] A. Forsgren. Inertia-controlling factorizations for optimization algorithms. *Appl. Num. Math.*, 43:91–107, 2002.

[72] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. Optim.*, 8:1132–1152, 1998.

[73] A. Forsgren, P. E. Gill, and W. Murray. On the identification of local minimizers in inertia-controlling methods for quadratic programming. *SIAM J. Matrix Anal. Appl.*, 12:730–746, 1991.

[74] M. P. Friedlander. *A Globally Convergent Linearly Constrained Lagrangian Method for Nonlinear Optimization*. PhD thesis, Department of Operations Research, Stanford University, Stanford, CA, 2002.

[75] M. P. Friedlander and S. Leyffer. Global and finite termination of a two-phase augmented Lagrangian filter method for general quadratic programs. *SIAM J. Sci. Comput.*, 30(4):1706–1729, 2008.

[76] M. P. Friedlander and M. A. Saunders. A globally convergent linearly constrained Lagrangian method for nonlinear optimization. *SIAM J. Optim.*, 15(3):863–897, 2005.

[77] M. P. Friedlander and P. Tseng. Exact regularization of convex programs. *SIAM J. Optim.*, 18(4):1326–1350, 2007.

[78] J. C. Gilbert and C. Lemaréchal. Some numerical experiments with variable-storage quasi-Newton algorithms. *Math. Program.*, pages 407–435, 1989.

[79] J. R. Gilbert and M. T. Heath. Computing a sparse basis for the null space. Report TR86-730, Department of Computer Science, Cornell University, 1986.

[80] P. E. Gill, N. I. M. Gould, W. Murray, M. A. Saunders, and M. H. Wright. A weighted Gram-Schmidt method for convex quadratic programming. *Math. Program.*, 30:176–195, 1984.

[81] P. E. Gill and M. W. Leonard. Limited-memory reduced-Hessian methods for large-scale unconstrained optimization. *SIAM J. Optim.*, 14:380–401, 2003.

[82] P. E. Gill and W. Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Program.*, 7:311–350, 1974.

[83] P. E. Gill and W. Murray. Numerically stable methods for quadratic programming. *Math. Program.*, 14:349–372, 1978.

[84] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Solving reduced KKT systems in barrier methods for linear programming. In G. A. Watson and D. Griffiths, editors, *Numerical Analysis 1993 (Dundee, 1993)*, volume 303 of *Pitman Res. Notes Math. Ser.*, pages 89–104. Longman Sci. Tech., Harlow, UK, 1994.

[85] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47:99–131, 2005.

[86] P. E. Gill, W. Murray, and M. A. Saunders. User's guide for SNOPT Version 7: Software for large-scale nonlinear programming. Numerical Analysis Report 06-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2006.

[87] P. E. Gill, W. Murray, M. A. Saunders, G. W. Stewart, and M. H. Wright. Properties of a representation of a basis for the null space. *Math. Programming*, 33(2):172–186, 1985.

[88] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. A note on a sufficient-decrease criterion for a nonderivative step-length procedure. *Math. Programming*, 23(3):349–352, 1982.

[89] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Procedures for optimization problems with a mixture of bounds and general linear constraints. *ACM Trans. Math. Software*, 10:282–298, 1984.

[90] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Sparse matrix methods in optimization. *SIAM J. Sci. Statist. Comput.*, 5(3):562–589, 1984.

[91] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Maintaining *LU* factors of a general sparse matrix. *Linear Algebra Appl.*, 88/89:239–270, 1987.

[92] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. A Schur-complement method for sparse quadratic programming. Report SOL 87-12, Department of Operations Research, Stanford University, Stanford, CA, 1987.

[93] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. A Schur-complement method for sparse quadratic programming. In M. G. Cox and S. J. Hammarling, editors, *Reliable Numerical Computation*, pages 113–138. Oxford University Press, 1990.

[94] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Inertia-controlling methods for general quadratic programming. *SIAM Rev.*, 33(1):1–36, 1991.

[95] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Some theoretical properties of an augmented Lagrangian merit function. In P. M. Pardalos, editor, *Advances in Optimization and Parallel Computing*, pages 101–128. North Holland, North Holland, 1992.

[96] P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1981.

[97] P. E. Gill and D. P. Robinson. A primal-dual augmented Lagrangian. *Computational Optimization and Applications*, pages 1–25, 2010.

[98] P. E. Gill and E. Wong. Methods for convex and general quadratic programming. Numerical Analysis Report 10-1, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2010.

[99] P. E. Gill and E. Wong. A regularized method for convex and general quadratic programming. Numerical Analysis Report 11-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2011.

[100] D. Goldfarb. A family of variable metric methods derived by variational means. *Math. Comp.*, 24:23–26, 1970.

[101] D. Goldfarb. Curvilinear path steplength algorithms for minimization which use directions of negative curvature. *Math. Program.*, 18:31–40, 1980.

[102] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Math. Programming*, 27(1):1–33, 1983.

[103] N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Math. Program.*, 32:90–99, 1985.

[104] N. I. M. Gould. On the accurate determination of search directions for simple differentiable penalty functions. *IMA J. Numer. Anal.*, 6:357–372, 1986.

[105] N. I. M. Gould. An algorithm for large-scale quadratic programming. *IMA J. Numer. Anal.*, 11(3):299–324, 1991.

[106] N. I. M. Gould, D. Orban, and P. L. Toint. *GALAHAD*, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Trans. Math. Software*, 29(4):353–372, Dec 2003.

[107] N. I. M. Gould and D. P. Robinson. A second derivative SQP method with imposed descent. Numerical Analysis Report 08/09, Computational Laboratory, University of Oxford, Oxford, UK, 2008.

[108] N. I. M. Gould and D. P. Robinson. A second derivative SQP method: Global convergence. *SIAM J. Optim.*, 20(4):2023–2048, 2010.

[109] N. I. M. Gould and D. P. Robinson. A second derivative SQP method: Local convergence and practical issues. *SIAM J. Optim.*, 20(4):2049–2079, 2010.

[110] N. I. M. Gould, J. A. Scott, and Y. Hu. A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. *ACM Trans. Math. Software*, 33(2):Art. 10, 32, 2007.

[111] N. I. M. Gould and P. L. Toint. An iterative working-set method for large-scale nonconvex quadratic programming. *Appl. Numer. Math.*, 43(1-2):109–128, 2002. 19th Dundee Biennial Conference on Numerical Analysis (2001).

[112] N. I. M. Gould and P. L. Toint. Numerical methods for large-scale non-convex quadratic programming. In *Trends in industrial and applied mathematics (Amritsar, 2001)*, volume 72 of *Appl. Optim.*, pages 149–179. Kluwer Acad. Publ., Dordrecht, 2002.

[113] J.-P. Goux and S. Leyffer. Solving large MINLPs on computational grids. *Optim. Eng.*, 3(3):327–346, 2002. Special issue on mixed-integer programming and its applications to engineering.

[114] J. Greenstadt. On the relative efficiencies of gradient methods. *Math. Comp.*, 21:360–367, 1967.

[115] L. Grippo, F. Lampariello, and S. Lucidi. Newton-type algorithms with nonmonotone line search for large-scale unconstrained optimization. In *System modelling and optimization (Tokyo, 1987)*, volume 113 of *Lecture Notes in Control and Inform. Sci.*, pages 187–196. Springer, Berlin, 1988.

[116] L. Grippo, F. Lampariello, and S. Lucidi. A truncated Newton method with nonmonotone line search for unconstrained optimization. *J. Optim. Theory Appl.*, 60(3):401–419, 1989.

[117] L. Grippo, F. Lampariello, and S. Lucidi. A class of nonmonotone stabilization methods in unconstrained optimization. *Numer. Math.*, 59(8):779–805, 1991.

[118] N.-z. Gu and J.-t. Mo. Incorporating nonmonotone strategies into the trust region method for unconstrained optimization. *Comput. Math. Appl.*, 55(9):2158–2172, 2008.

[119] W. W. Hager. Stabilized sequential quadratic programming. *Comput. Optim. Appl.*, 12(1-3):253–273, 1999. Computational optimization—a tribute to Olvi Mangasarian, Part I.

[120] S. P. Han. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Math. Programming*, 11(3):263–282, 1976/77.

[121] S. P. Han. A globally convergent method for nonlinear programming. *J. Optim. Theory Appl.*, 22:297–309, 1977.

[122] S. P. Han and O. L. Mangasarian. Exact penalty functions in nonlinear programming. *Math. Programming*, 17(3):251–269, 1979.

[123] J. Herskovits. A two-stage feasible directions algorithm for nonlinear constrained optimization. *Math. Programming*, 36(1):19–38, 1986.

[124] M. R. Hestenes. Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4:303–320, 1969.

[125] H. M. Huynh. *A Large-Scale Quadratic Programming Solver Based on Block-LU Updates of the KKT System.* PhD thesis, Program in Scientific Computing and Computational Mathematics, Stanford University, Stanford, CA, 2008.

[126] M. M. Kostreva and X. Chen. A superlinearly convergent method of feasible directions. *Appl. Math. Comput.*, 116(3):231–244, 2000.

[127] M. M. Kostreva and X. Chen. Asymptotic rates of convergence of SQP-type methods of feasible directions. In *Optimization methods and applications*, volume 52 of *Appl. Optim.*, pages 247–265. Kluwer Acad. Publ., Dordrecht, 2001.

[128] J. Kroyan. *Trust-Search Algorithms for Unconstrained Optimization.* PhD thesis, Department of Mathematics, University of California, San Diego, February 2004.

[129] C. T. Lawrence and A. L. Tits. A computationally efficient feasible sequential quadratic programming algorithm. *SIAM J. Optim.*, 11(4):1092–1118 (electronic), 2001.

[130] S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Comput. Optim. Appl.*, 18(3):295–309, 2001.

[131] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45:503–528, 1989.

[132] X.-W. Liu and Y.-X. Yuan. A robust algorithm for optimization with general equality and inequality constraints. *SIAM J. Sci. Comput.*, 22(2):517–534 (electronic), 2000.

[133] C. M. Maes. *A Regularized Active-Set Method for Sparse Convex Quadratic Programming.* PhD thesis, Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, August 2010.

[134] A. Majthay. Optimality conditions for quadratic programming. *Math. Programming*, 1:359–365, 1971.

[135] O. L. Mangasarian and S. Fromovitz. The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *J. Math. Anal. Appl.*, 17:37–47, 1967.

[136] N. Maratos. *Exact Penalty Function Algorithms for Finite-Dimensional and Control Optimization Problems.* PhD thesis, Department of Computing and Control, University of London, 1978.

[137] J. Mo, K. Zhang, and Z. Wei. A variant of SQP method for inequality constrained optimization and its global convergence. *J. Comput. Appl. Math.*, 197(1):270–281, 2006.

[138] J. L. Morales. A numerical study of limited memory BFGS methods. *Appl. Math. Lett.*, 15(4):481–487, 2002.

[139] J. L. Morales, J. Nocedal, and Y. Wu. A sequential quadratic programming algorithm with an additional equality constrained phase. Technical Report OTC-05, Northwestern University, 2008.

[140] J. J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Math. Program.*, 16:1–20, 1979.

[141] J. J. Moré and D. C. Sorensen. Newton's method. In G. H. Golub, editor, *Studies in Mathematics, Volume 24. MAA Studies in Numerical Analysis*, pages 29–82. Math. Assoc. America, Washington, DC, 1984.

[142] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Software*, 20(3):286–307, 1994.

[143] W. Murray. An algorithm for constrained minimization. In *Optimization (Sympos., Univ. Keele, Keele, 1968)*, pages 247–258. Academic Press, London, 1969.

[144] W. Murray and F. J. Prieto. A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM J. Optim.*, 5:590–640, 1995.

[145] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer-Verlag, New York, 1999.

[146] A. Olivares, J. M. Moguerza, and F. J. Prieto. Nonconvex optimization using negative curvature within a modified linesearch. *European J. Oper. Res.*, 189(3):706–722, 2008.

[147] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Academic Press, New York, 1970.

[148] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1970 original.

[149] E. R. Panier and A. L. Tits. A superlinearly convergent feasible method for the solution of inequality constrained optimization problems. *SIAM J. Control Optim.*, 25(4):934–950, 1987.

[150] E. R. Panier and A. L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Math. Programming*, 59(2, Ser. A):261–276, 1993.

[151] P. M. Pardalos and G. Schnitger. Checking local optimality in constrained quadratic programming is NP-hard. *Oper. Res. Lett.*, 7(1):33–35, 1988.

[152] P. M. Pardalos and S. A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *J. Global Optim.*, 1(1):15–22, 1991.

[153] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298, London and New York, 1969. Academic Press.

[154] M. J. D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In *Nonlinear Programming, 3 (Proc. Sympos., Special Interest Group Math. Programming, Univ. Wisconsin, Madison, Wis., 1977)*, pages 27–63. Academic Press, New York, 1978.

[155] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G. A. Watson, editor, *Numerical Analysis, Dundee 1977*, number 630 in Lecture Notes in Mathematics, pages 144–157, Heidelberg, Berlin, New York, 1978. Springer Verlag.

[156] M. J. D. Powell. On the quadratic programming algorithm of Goldfarb and Idnani. *Math. Programming Stud.*, (25):46–61, 1985.

[157] D. P. Robinson. *Primal-Dual Methods for Nonlinear Optimization.* PhD thesis, Department of Mathematics, University of California, San Diego, September 2007.

[158] S. M. Robinson. A quadratically-convergent algorithm for general nonlinear programming problems. *Math. Program.*, 3:145–156, 1972.

[159] S. M. Robinson. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms. *Math. Program.*, 7:1–16, 1974.

[160] R. T. Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM J. Control Optim.*, 12:268–285, 1974.

[161] M. A. Saunders. Cholesky-based methods for sparse least squares: The benefits of regularization. In L. Adams and J. L. Nazareth, editors, *Linear and Nonlinear Conjugate Gradient-Related Methods*, pages 92–100. SIAM Publications, Philadelphia, 1996. Proceedings of AMS-IMS-SIAM Joint Summer Research Conference, University of Washington, Seattle, WA (July 9–13, 1995).

[162] M. A. Saunders and J. A. Tomlin. Solving regularized linear programs using barrier methods and KKT systems. Report SOL 96-4, Dept of EESOR, Stanford University, 1996.

[163] M. A. Saunders and J. A. Tomlin. Stable reduction to KKT systems in barrier methods for linear and quadratic programming. Report SOL 96-3, Dept of EESOR, Stanford University, 1996.

[164] O. Schenk and K. Gärtner. Solving unsymmetric sparse systems of linear equations with PARDISO. In *Computational science—ICCS 2002, Part II (Amsterdam)*, volume 2330 of *Lecture Notes in Comput. Sci.*, pages 355–363. Springer, Berlin, 2002.

[165] K. Schittkowski. The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function. I. Convergence analysis. *Numer. Math.*, 38(1):83–114, 1981/82.

[166] K. Schittkowski. The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function. II. An efficient implementation with linear least squares subproblems. *Numer. Math.*, 38(1):115–127, 1981/82.

[167] K. Schittkowski. On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function. *Math. Operationsforsch. Statist. Ser. Optim.*, 14(2):197–216, 1983.

[168] R. B. Schnabel and E. Eskow. A new modified Cholesky factorization. *SIAM J. Sci. and Statist. Comput.*, 11:1136–1158, 1990.

[169] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.*, 24:647–656, 1970.

[170] R. A. Tapia. A stable approach to Newton's method for general mathematical programming problems in $\mathbb{R}^n$. *J. Optim. Theory Appl.*, 14:453–476, 1974.

[171] R. A. Tapia. Diagonalized multiplier methods and quasi-Newton methods for constrained optimization. *J. Optim. Theory Appl.*, 22:135–194, 1977.

[172] P. L. Toint. An assessment of nonmonotone linesearch techniques for unconstrained optimization. *SIAM J. Sci. Comput.*, 17(3):725–739, 1996.

[173] S. Ulbrich. On the superlinear local convergence of a filter-SQP method. *Math. Program.*, 100(1, Ser. B):217–245, 2004.

[174] G. Van der Hoek. Asymptotic properties of reduction methods applying linearly equality constrained reduced problems. *Math. Program.*, 16:162–189, 1982.

[175] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: local convergence. *SIAM J. Optim.*, 16(1):32–48 (electronic), 2005.

[176] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: motivation and global convergence. *SIAM J. Optim.*, 16(1):1–31 (electronic), 2005.

[177] R. B. Wilson. *A Simplicial Method for Convex Programming*. PhD thesis, Harvard University, 1963.

[178] S. J. Wright. Superlinear convergence of a stabilized SQP method to a degenerate solution. *Comput. Optim. Appl.*, 11(3):253–275, 1998.

[179] S. J. Wright. Modifying SQP for degenerate problems. *SIAM J. Optim.*, 13(2):470–497, 2002.

[180] S. J. Wright. An algorithm for degenerate nonlinear programming with rapid local convergence. *SIAM J. Optim.*, 15(3):673–696, 2005.

[181] Y.-X. Yuan. Conditions for convergence of trust region algorithms for nonsmooth optimization. *Math. Programming*, 31(2):220–228, 1985.

[182] Y.-X. Yuan. On the superlinear convergence of a trust region algorithm for nonsmooth optimization. *Math. Programming*, 31(3):269–285, 1985.

[183] Y.-X. Yuan. On the convergence of a new trust region algorithm. *Numer. Math.*, 70(4):515–539, 1995.

[184] W. I. Zangwill. Non-linear programming via penalty functions. *Management Sci.*, 13:344–358, 1967.

[185] H. Zhang and W. W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.*, 14(4):1043–1056 (electronic), 2004.