ELSEVIER

Journal of Computational and Applied Mathematics 124 (2000) 123–137

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

Sequential quadratic programming for large-scale nonlinear optimization ☆

Paul T. Boggs[a,*], Jon W. Tolle[b]

[a] *Computational Sciences and Mathematics Research Department, Sandia National Laboratories, Livermore, CA 94550, USA*

[b] *Departments of Operations Research and Mathematics, University of North Carolina, Chapel Hill, NC 27514, USA*

Received 12 July 1999; received in revised form 17 December 1999

Abstract

The sequential quadratic programming (SQP) algorithm has been one of the most successful general methods for solving nonlinear constrained optimization problems. We provide an introduction to the general method and show its relationship to recent developments in interior-point approaches, emphasizing large-scale aspects. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Sequential quadratic programming; Nonlinear optimization; Newton methods; Interior-point methods; Local; Trust-region methods convergence; Global convergence

1. Introduction

In this article we consider the general method of Sequential Quadratic Programming (SQP) for solving the nonlinear programming problem

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) \\ &\text{subject to} \quad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \\ &\qquad\qquad \boldsymbol{g}(\boldsymbol{x}) \leqslant \boldsymbol{0}, \end{aligned} \tag{NLP}$$

where $f : \mathscr{R}^n \to \mathscr{R}$, $\boldsymbol{h} : \mathscr{R}^n \to \mathscr{R}^m$, and $\boldsymbol{g} : \mathscr{R}^n \to \mathscr{R}^p$. Broadly defined, the SQP method is a procedure that generates iterates converging to a solution of this problem by solving quadratic programs that are approximations to (NLP). In its many implemented forms, this method has been shown to be a

☆ Contribution of Sandia National Laboratories and not subject to copyright in the United States.

* Corresponding author.

E-mail addresses: ptboggs@ca.sandia.gov (P.T. Boggs), tolle@email.unc.edu (J.W. Tolle).

0377-0427/00/$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.
PII: S0377-0427(00)00429-5

very useful tool for solving nonlinear programs, especially where a significant degree of nonlinearity is present. In this paper our goals are to provide a brief synopsis of the general method, to introduce some of the more recent results, and to provide direction for further investigation. As part of our exposition we relate SQP to the applications of interior-point methods to nonlinear programming. The discussion will be more motivational than rigorous, our emphasis being on the exposition of underlying issues and ideas rather than on detailed theorems and implementation techniques. We will, for the most part, discuss the algorithm and its properties without making any special assumptions about the structure of the problem. While the SQP algorithm is applicable to all sizes of nonlinear programming problems, problems of large scale (i.e., a large number of variables and/or constraints) are the most challenging and therefore the ones where the development of efficient strategies for their solution will have the most impact. Accordingly, our presentation will be slanted towards the procedures which are likely to prove useful for solving large problems. We will provide explicit references for the recent theoretical and computational results, but the literature for SQP is immense and it is beyond the scope of this paper to do it justice. Instead we direct the reader to [5] for a more comprehensive list of sources.

An outline of the paper is as follows: a particular basic formulation of the nonlinear program and its corresponding necessary conditions will be presented in Section 2 followed, in Section 3, by examples of quadratic programming approximations; aspects of the local and global convergence theory will be provided in Sections 4 and 5; in Section 6 various important issues in the solution of the quadratic subproblem will be discussed; and, finally, in Section 7 the important ideas of reduced Hessian SQP methods are presented. A knowledge of basic optimization theory and practice (for example, as developed in [17]) is adequate for following the ideas contained herein.

2. The formulation of the necessary conditions for NLP

We begin by introducing some terminology and notation that is necessary to describe the method; additional terminology and notation will be introduced as it is needed.

Throughout the paper we use bold face letters to represent vectors (both variables and functions) and plain face for scalars and matrices. The subscript or superscript k is used to indicate a kth iterate; similarly an asterisk indicates an optimal solution (or multiplier). We use ∇ to indicate the derivative of a (scalar or vector-valued) function and $\mathscr{H}$ to indicate the Hessian of a scalar function. Sometimes subscripts will be added to indicate the variables with respect to which differentiation is performed; if no subscript is present the differentiation is assumed to be with respect to the vector $\boldsymbol{x}$ only. Unless specified otherwise all norms are assumed to be the Euclidean norm for a vector and the induced operator norm for matrices. We will use the superscript "t" to indicate the transpose of a vector or matrix. Finally, the symbol $\boldsymbol{x} \odot \boldsymbol{y}$ will denote the vector defined by componentwise multiplication of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$.

Associated with (NLP) is the *standard Lagrangian function*

$$L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}}\boldsymbol{u} + \boldsymbol{g}(\boldsymbol{x})^{\mathrm{t}}\boldsymbol{v},$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the multiplier (dual) vectors. We denote the active set of inequality constraints at $\boldsymbol{x}$ by $\mathscr{A}(\boldsymbol{x})$, i.e., $\mathscr{A}(\boldsymbol{x}) = \{i\colon \boldsymbol{g}_i(\boldsymbol{x}) = 0\}$, and by $D(\boldsymbol{x})$ the $n \times (m + |\mathscr{A}(\boldsymbol{x})|)$ matrix whose columns

are the gradients (with respect to $\boldsymbol{x}$) of the equality and active inequality constraints at $\boldsymbol{x}$. We make the following assumptions concerning (NLP):

A.1 All of the functions in the nonlinear program have Lipschitz continuous second derivatives.

A.2 (NLP) has a feasible (local) solution $\boldsymbol{x}^*$ with optimal multiplier vectors $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ satisfying the first order conditions

$$\nabla_x L(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*) = \boldsymbol{0},$$
$$\boldsymbol{g}(\boldsymbol{x}^*) \odot \boldsymbol{v}^* = \boldsymbol{0},$$
$$\boldsymbol{v}^* \geqslant \boldsymbol{0}.$$

A.3 Strict complementary slackness ($i \in \mathscr{A}(\boldsymbol{x}^*)$ implies $v_i^* > 0$) holds.

A.4 The matrix $D(\boldsymbol{x}^*)$ has full column rank.

A.5 The second-order sufficiency condition holds, i.e., for all $\boldsymbol{y} \neq \boldsymbol{0}$ such that $D(\boldsymbol{x}^*)^{\mathrm{t}}\boldsymbol{y} = \boldsymbol{0}$ the strict inequality $\boldsymbol{y}^{\mathrm{t}} \mathscr{H} L_* \boldsymbol{y} > 0$ is valid, where $\mathscr{H} L_*$ is the Hessian of the Lagrangian with respect to $\boldsymbol{x}$ evaluated at $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$.

For the analysis that follows it is advantageous to add the vector of slack variables, $\boldsymbol{z}$, and put the general nonlinear programming problem into slack variable form

$$\begin{aligned} \underset{(\boldsymbol{x},\boldsymbol{z})}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \\ & \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{z} = \boldsymbol{0}, \\ & \boldsymbol{z} \geqslant \boldsymbol{0}. \end{aligned} \tag{NLP}$$

In this form the first-order necessary and feasibility conditions that a solution $(\boldsymbol{x}^*, \boldsymbol{z}^*)$ and its multipliers $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ must satisfy are the following:

$$\nabla f(\boldsymbol{x}^*) + \nabla \boldsymbol{h}(\boldsymbol{x}^*)\boldsymbol{u}^* + \nabla \boldsymbol{g}(\boldsymbol{x}^*)\boldsymbol{v}^* = \boldsymbol{0}, \tag{1}$$

$$\boldsymbol{h}(\boldsymbol{x}^*) = \boldsymbol{0}, \tag{2}$$

$$\boldsymbol{g}(\boldsymbol{x}^*) + \boldsymbol{z}^* = \boldsymbol{0}, \tag{3}$$

$$\boldsymbol{z}^* \odot \boldsymbol{v}^* = \boldsymbol{0}, \tag{4}$$

$$\boldsymbol{z}^* \geqslant \boldsymbol{0}, \tag{5}$$

$$\boldsymbol{v}^* \geqslant \boldsymbol{0}. \tag{6}$$

Defining the *extended Lagrangian function*

$$\mathscr{L}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}}\boldsymbol{u} + (\boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{z})^{\mathrm{t}}\boldsymbol{v},$$

we see that solving this version of (NLP) is equivalent to solving the problem

$$\begin{aligned} \underset{(\boldsymbol{x},\boldsymbol{z})}{\text{minimize}} \quad & \mathscr{L}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}) \\ \text{subject to} \quad & \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \\ & \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{z} = \boldsymbol{0}, \\ & \boldsymbol{z} \geqslant 0 \end{aligned} \tag{LNLP}$$

for some $\boldsymbol{u}$ and some $\boldsymbol{v} \geqslant \boldsymbol{0}$ satisfying the complementary slackness conditions $\boldsymbol{z} \odot \boldsymbol{v} = \boldsymbol{0}$.

The fundamental approach of the SQP method is to solve (NLP) by solving a sequence of quadratic programs that are approximations to (LNLP). In the next section we will provide some of the more common quadratic approximations used in SQP.

3. Examples of quadratic subproblems

The first example, the standard approximation for the basic SQP method, is the Taylor Series approximation to the problem (LNLP). Given an estimate of the variables, $(\boldsymbol{x}^k, \boldsymbol{z}^k)$, and the multiplier vectors, $(\boldsymbol{u}^k, \boldsymbol{v}^k)$, the quadratic program approximation generated by the Taylor Series for a change, $(\boldsymbol{d}_x, \boldsymbol{d}_z)$, in the vectors $\boldsymbol{x}^k$ and $\boldsymbol{z}^k$ is given by

$$\begin{aligned}
&\underset{(\boldsymbol{d}_x,\boldsymbol{d}_z)}{\text{minimize}} \ \tfrac{1}{2}\boldsymbol{d}_x^{\mathrm{t}}\mathscr{H}\mathscr{L}_k\boldsymbol{d}_x + (\nabla_x\mathscr{L}_k)^{\mathrm{t}}\boldsymbol{d}_x + (\nabla_x\mathscr{L}_k)^{\mathrm{t}}\boldsymbol{d}_z \\
&\text{subject to} \quad \nabla\boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x = -\boldsymbol{h}(\boldsymbol{x}^k), \\
&\qquad\qquad \nabla\boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \boldsymbol{d}_z = -(\boldsymbol{g}(\boldsymbol{x}^k) + \boldsymbol{z}^k), \\
&\qquad\qquad \boldsymbol{d}_z \geqslant -\boldsymbol{z}^k,
\end{aligned}$$

where the subscript on $\mathscr{L}$ indicates that the derivatives are evaluated at $(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)$. Some simplification of this quadratic program is possible since, when the constraints are satisfied, the gradient of the Lagrangian terms reduce to $\nabla f(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x$. In addition, $\mathscr{H}\mathscr{L}_k$ is the same as $\mathscr{H}L_k$. The purpose of using the Lagrangian function in the objective function is now clear; even though the constraints are linearized in forming the approximating quadratic program, second order information on the constraint functions is maintained in the objective function via the Hessian of the Lagrangian. In many situations, this Hessian matrix is either unavailable or too costly to evaluate. In these cases a finite difference approximation or a *Quasi-Newton update*, i.e., a matrix that depends on first order information at the preceding iterates, may be used in place of the Hessian. In the latter case, the Hessian approximation is taken to be a positive definite matrix, which makes solving the quadratic subproblem easier. Representing the true Hessian or an approximation thereof by B_k, we can now rewrite the approximating quadratic program as

$$\begin{aligned}
&\underset{(\boldsymbol{d}_x,\boldsymbol{d}_z)}{\text{minimize}} \ \tfrac{1}{2}\boldsymbol{d}_x^{\mathrm{t}}B_k\boldsymbol{d}_x + \nabla f(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x \\
&\text{subject to} \quad \nabla\boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x = -\boldsymbol{h}(\boldsymbol{x}^k), \\
&\qquad\qquad \nabla\boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \boldsymbol{d}_z = -(\boldsymbol{g}(\boldsymbol{x}^k) + z^k), \\
&\qquad\qquad \boldsymbol{d}_z \geqslant -\boldsymbol{z}^k.
\end{aligned} \tag{QP1}$$

We note that if the variable $\boldsymbol{d}_z$ is eliminated this quadratic program is a quadratic approximation to the original nonslack form of (NLP).

As a second example of an approximating quadratic program we cast a version of the nonlinear interior-point algorithm into this framework. In this approach, the nonnegative slack variable constraint in (LNLP) is put into the objective function as a log barrier function so that the problem has

the form

$$\begin{aligned} \underset{(\boldsymbol{x},\boldsymbol{z})}{\text{minimize}} \quad & \mathscr{L}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{u},\boldsymbol{v}) - \rho\sum_{j=1}^{p}(\log(\boldsymbol{z}_j)) \\ \text{subject to} \quad & \boldsymbol{h}(\boldsymbol{x}) = 0, \\ & \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{z} = 0, \end{aligned} \tag{NLPI(ρ))}$$

where ρ is a positive barrier parameter that is chosen to tend to zero in an appropriate manner. The nonnegativity constraint on $\boldsymbol{z}$ is implicitly enforced by the log function. The corresponding quadratic approximation is

$$\begin{aligned} \underset{(\boldsymbol{d}_x,\boldsymbol{d}_z)}{\text{minimize}} \quad & \tfrac{1}{2}\boldsymbol{d}_x^{\mathrm{t}}B_k\boldsymbol{d}_x + \tfrac{1}{2}\rho\boldsymbol{d}_z^{\mathrm{t}}Z_k^{-2}\boldsymbol{d}_z + \nabla f(x^k)^{\mathrm{t}}\boldsymbol{d}_x - \rho\mathbf{e}^{\mathrm{t}}Z_k^{-1}\boldsymbol{d}_z \\ \text{subject to} \quad & \nabla\boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x = -\boldsymbol{h}(\boldsymbol{x}^k), \\ & \nabla\boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \boldsymbol{d}_z = -\boldsymbol{g}(\boldsymbol{x}^k + \boldsymbol{z}^k), \end{aligned} \tag{QP2(ρ))}$$

where $\mathbf{e}$ is the vector of ones, Z_k is the diagonal matrix with components of $\boldsymbol{z}^k$ on the diagonal, and the matrix B_k represents either the Hessian of L or its approximation. The interpretation of the interior-point methods in terms of SQP algorithms is not conventional, but as will be seen it fits naturally into that context in terms of the local convergence analysis. A general interior-point algorithm for solving (NLP) has recently been given in [19].

As a final example we mention a version of an SQP method employing a trust region. In this approach, a constraint limiting the size of the step is included in the constraints. Thus the quadratic subproblem has the form of either (QP1) or (QP2(ρ)) with the added constraint

$$||(\boldsymbol{d}_x,\boldsymbol{d}_z)|| \leqslant \tau_k. \tag{7}$$

Here, as in trust-region algorithms for unconstrained optimization, τ_k is a positive parameter that measures the adequacy of the quadratic approximation to the original problem. A recent work using this approach is [7].

In the implementation of each of these methods, the particular quadratic programs are (approximately) solved to obtain $(\boldsymbol{d}_x,\boldsymbol{d}_z)$. These steps can then be used to compute $\boldsymbol{x}^{k+1}$ and $\boldsymbol{z}^{k+1}$ by

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \boldsymbol{x}^k + \alpha\boldsymbol{d}_x, \\ \boldsymbol{z}^{k+1} &= \boldsymbol{z}^k + \alpha\boldsymbol{d}_z, \end{aligned}$$

where α is a step length parameter that may be used to assure the nonnegativity of $\boldsymbol{z}^{k+1}$ and global convergence (see Section 5). In the case where a trust-region constraint is included, the control of the step length is implicitly included in the solution. Updates, $\boldsymbol{u}^{k+1}$ and $\boldsymbol{v}^{k+1}$, for the multipliers can be determined directly or obtained from the solution of the quadratic programs. We note that there are many methods for solving quadratic programs; for example, active set methods, interior-point methods, and reduced Hessian techniques have all been used in SQP algorithms. Some methods do not lead to estimates of the multiplier vectors directly and so methods for accurately estimating these vectors need to be provided. In Section 6 we discuss some of the methods that are used to solve the quadratic subproblems.

The analysis of the SQP methods can be divided into two distinct, but related, parts: local convergence, which is concerned with the asymptotic rate of the convergence, and global convergence,

which concerns the convergence of the iterates when the initial point is not close to the solution. These two issues will be considered in Sections 4 and 5.

4. Local convergence

In this section we discuss the local convergence properties of the SQP method. By local convergence we mean that the algorithm will generate a sequence of iterates that converges to an optimal solution-multiplier vector provided that the initial iterate is sufficiently close to that optimal solution. Associated with this convergence is the asymptotic *rate of convergence* which indicates the rapidity with which the discrepancy between the iterates and the solution goes to zero. Any local convergence results will depend on the details of the implementation of the SQP algorithm and, in particular, on how accurately the quadratic programs are solved. Rather than provide results for any specific implementation, we will use the necessary conditions for the approximating quadratic programs to relate their solutions to the steps taken by Newton's method for solving Eqs. (1)–(6) as described below. Although only locally applicable, Newton's method provides a conventional standard which can be used to measure the asymptotic convergence properties of any particular SQP method. Toward this end, we observe that given a good approximation, $(\boldsymbol{x}^k, \boldsymbol{z}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)$, to the optimal solution and the optimal multiplier vectors of (NLP), Newton's method requires solving the linear system

$$\begin{bmatrix} \mathscr{H}L_k & 0 & \nabla \boldsymbol{h}(\boldsymbol{x}^k) & \nabla \boldsymbol{g}(\boldsymbol{x}^k) \\ \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} & 0 & 0 & 0 \\ \nabla \boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}} & I & 0 & 0 \\ 0 & V_k & 0 & Z_k \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_x \\ \boldsymbol{d}_z \\ \boldsymbol{d}_u \\ \boldsymbol{d}_v \end{bmatrix} = \begin{bmatrix} -\nabla_x L_k \\ -\boldsymbol{h}(\boldsymbol{x}^k) \\ -\boldsymbol{g}(\boldsymbol{x}^k) - \boldsymbol{z}^k \\ -Z_k \boldsymbol{v}^k \end{bmatrix} \tag{8}$$

for $(\boldsymbol{d}_x, \boldsymbol{d}_z, \boldsymbol{d}_u, \boldsymbol{d}_v)$ where V_k and Z_k are diagonal matrices whose diagonals are $\boldsymbol{x}^k$ and $\boldsymbol{z}^k$. The next iterate is then given by

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \boldsymbol{x}^k + \boldsymbol{d}_x, \\ \boldsymbol{z}^{k+1} &= \boldsymbol{z}^k + \boldsymbol{d}_z, \\ \boldsymbol{u}^{k+1} &= \boldsymbol{u}^k + \boldsymbol{d}_u, \\ \boldsymbol{v}^{k+1} &= \boldsymbol{v}^k + \boldsymbol{d}_v. \end{aligned} \tag{9}$$

Under our basic assumptions (A.1–A.5) the coefficient matrix for (8) is nonsingular in a neighborhood of the solution and hence Newton's method is well defined and converges quadratically. The nonnegativity restrictions on the slack vector and the inequality multiplier vector are not required to be explicitly enforced because of the local convergence properties of Newton's method.

We first observe that if the B_k are taken as the true Hessians, $\mathscr{H}L_k$, the solutions of the quadratic subproblems (QP1) and (QP2(ρ)) lead to approximations of the Newton step, differing only in the approximation of the complementary slackness conditions. (We note that the local analyses of the trust-region methods generally reduce to that of these two methods since it is assumed that the added constraint will be strictly satisfied as the solution is approached.) As above, we assume that an approximate solution and its corresponding multiplier vectors, $(\boldsymbol{x}^k, \boldsymbol{z}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)$, are known. We

denote by $(\boldsymbol{d}_x, \boldsymbol{d}_z, \boldsymbol{u}_{qp}, \boldsymbol{v}_{qp})$ the optimal solutions and multiplier of the quadratic problems (QP1) or (QP2(ρ)). Setting

$$\begin{aligned} \boldsymbol{d}_u &= \boldsymbol{u}_{qp} - \boldsymbol{u}^k, \\ \boldsymbol{d}_v &= \boldsymbol{v}_{qp} - \boldsymbol{v}_k, \end{aligned} \tag{10}$$

the first-order and feasibility conditions for the quadratic programs lead to the first three sets of equations in system (8) together with a fourth set which depends on the particular quadratic approximation. For (QP1) the complementary slackness conditions lead to

$$V_k \boldsymbol{d}_z + Z_k \boldsymbol{d}_v + \boldsymbol{d}_z \odot \boldsymbol{d}_v = -Z_k \boldsymbol{v}^k, \tag{11}$$

while the Lagrangian condition in $\boldsymbol{d}_z$ for (QP2(ρ)) leads to

$$\rho Z_k^{-1} \boldsymbol{d}_z + Z_k \boldsymbol{d}_v = \rho \mathbf{e} - Z_k \boldsymbol{v}^k. \tag{12}$$

Either of these last systems of equations may be taken as a perturbation of the linearized complementary slackness condition in (8) above, i.e., they can be written in the form

$$V_k \boldsymbol{d}_z + Z_k \boldsymbol{d}_v = -Z_k \boldsymbol{v}^k + \mathbf{p}(\boldsymbol{x}^k, \boldsymbol{z}^k, \boldsymbol{v}^k, \boldsymbol{d}_x, \boldsymbol{d}_z, \boldsymbol{d}_v), \tag{13}$$

where $\mathbf{p}$ is some perturbation function. It is worthwhile to point out that a linearization of the complementary slackness condition of the form

$$V_k \boldsymbol{d}_z + Z_k \boldsymbol{d}_v = \rho \mathbf{e} - Z_k \boldsymbol{v}^k \tag{14}$$

is used in place of (12) in most interior-point algorithms. This can also be written in the form (13).

In any case, if the vector $(\boldsymbol{d}_x, \boldsymbol{d}_z, \boldsymbol{u}_{\mathrm{qp}}, \boldsymbol{v}_{\mathrm{qp}})$ is obtained by solving one of the quadratic subproblems (QP1) or (QP2(ρ)), the vectors $\boldsymbol{d}_u$ and $\boldsymbol{d}_v$ are defined by (10), and new iterates $(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{v}^{k+1})$ are given by (9), then we have a canonical form of an SQP algorithm. The fact that these iterates approximate the Newton iterates provides the underlying motivation for using the SQP method.

If the vector $(\boldsymbol{d}_x, \boldsymbol{d}_z, \boldsymbol{d}_u, \boldsymbol{d}_v)$ is determined by *exactly* solving the particular first-order conditions for the quadratic subproblems (with $B_k = \mathscr{H} L_k$) and the new iterates are obtained using (9), the local convergence theory for SQP methods can be analyzed in terms of a perturbation of the Newton method for solving (1)–(4). Specifically, the results depend on how well the complementary slackness conditions in (8) are approximated by a particular choice of (11), (12), or (14). If there are no inequality constraints the iterates are identical to the Newton iterates for solving (1) and (2). Hence if the initial solution vector $\boldsymbol{x}^0$ and initial multiplier vector $\boldsymbol{u}^0$ are sufficiently close to the optimal solution and multiplier vector then exactly solving the quadratic programs leads to the quadratic convergence of the iterates to the optimal solution-multiplier vector. Note that the initial multiplier vector $\boldsymbol{u}^0$ can always be taken as the least squares solution to the Lagrangian condition when $\boldsymbol{x}^0$ is sufficiently close to the optimal solution.

If inequality constraints are present, a local convergence analysis can be given in the case where the quadratic problem (QP1) is solved exactly by utilizing the fact that when the iterates are sufficiently close to the optimal solution-multiplier vector the active sets at optimality and for (QP1) are the same. Thus the problem essentially reduces to an equality-constrained problem at that point and the Newton theory applies. Quadratic convergence can also be obtained in the interior-point scheme provided that the parameter ρ is chosen appropriately (relevant references can be found in [16]).

From a practical point of view, the local convergence analysis depends on many factors other than the form of the approximation of the complementary slackness conditions; it also depends on

the details that determine the specific implementation of an SQP method such as the accuracy of the solution of the quadratic approximation and how the step length parameter is chosen. Here we provide general local convergence results based on the perturbation method described above and then discuss how possible implementations fit (or do not fit) into this scheme. The basic procedure can be described as follows:

- Let the solutions and multipliers to the quadratic subproblem satisfy the system

$$A_k \mathbf{d}_w = -\mathbf{a}^k + \mathbf{p}(\mathbf{w}^k, \mathbf{d}_w). \tag{15}$$

where

$$\mathbf{w}^k = \begin{bmatrix} \boldsymbol{x}^k \\ \boldsymbol{z}^k \\ \boldsymbol{u}^k \\ \boldsymbol{v}^k \end{bmatrix}, \qquad \mathbf{d}_w = \begin{bmatrix} \boldsymbol{d}_x \\ \boldsymbol{d}_z \\ \boldsymbol{d}_u \\ \boldsymbol{d}_v \end{bmatrix}, \qquad A_k = \begin{bmatrix} B_k & 0 & \nabla \boldsymbol{h}(\boldsymbol{x}^k) & \nabla \boldsymbol{g}(\boldsymbol{x}^k) \\ \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} & 0 & 0 & 0 \\ \nabla \boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}} & I & 0 & 0 \\ 0 & V_k & 0 & Z_k \end{bmatrix},$$

$$\mathbf{a}^k = - \begin{bmatrix} \nabla_{\boldsymbol{x}} L_k \\ \boldsymbol{h}(\boldsymbol{x}^k) \\ \boldsymbol{g}(\boldsymbol{x}^k) + \boldsymbol{z}^k \\ V_k \boldsymbol{z}^k \end{bmatrix},$$

and $\mathbf{p}$ is a perturbation function that reflects the different approximations of the complementary slackness condition. Differences due to a nonexact solution of the quadratic programs can also be included in this term.

- Update the iterates according to $\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha_k \mathbf{d}_w$ for some $\alpha_k > 0$.

The local convergence of the iterates can then be analyzed by comparing them to the Newton iterates obtained from (8) and (9). The analysis depends on the size of $\mathbf{p}(\mathbf{w}^k, \mathbf{d}_w)$, the values of α_k, and how well B_k approximates $\mathscr{H} L_k$. More complexity is introduced by the fact that the components of $\mathbf{w}^k$ have different local convergence rates. Risking the possibility of oversimplification we will restrict our attention to the convergence rates of the primal variables $\boldsymbol{x}^k$ and $\boldsymbol{z}^k$. Generally, the multiplier vectors converge at a slower rate than the primal variables.

For the Hessian matrix approximations we make the following assumptions:

B.1 For each k the matrix B_k satisfies the conditions that $\boldsymbol{y} \neq \mathbf{0}$ and $D(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{y} = \mathbf{0}$ imply that $\boldsymbol{y}^{\mathrm{t}} B_k \boldsymbol{y} > 0$.

B.2 There exist constant η_1 and η_2 independent of k such that

$$||B_{k+1} - \mathscr{H} L_k|| \leqslant (1 + \eta_1 \sigma_k)||B_k - \mathscr{H} L_k|| + \eta_2 \sigma_k,$$

where

$$\sigma_k = \mathrm{O}(||\mathbf{w}^{k+1} - \mathbf{w}^*|| + ||\mathbf{w}^k - \mathbf{w}^*||).$$

Condition B.1 guarantees that the quadratic subproblem with B_k replacing $\mathscr{H} L_k$ has a solution. Condition B.2 is a *bounded deterioration* property on the sequence of matrices that is a common assumption for quasi-Newton methods in solving nonlinear systems and optimization problems. It ensures that the approximations B_k do not wander too far from the true Hessian $\mathscr{H} L_*$ at the solution.

A basic convergence theorem can now be stated. (These are not the weakest conditions under which the results are valid, but they are satisfactory for many, if not most purposes.)

Theorem 4.1. *Suppose that the following conditions are satisfied*:
(i) $\lim_{k\to\infty} \alpha_k = 1$;
(ii) $\mathbf{p}(\mathbf{w}^k, \mathbf{d}_w) = \mathbf{o}(||\mathbf{w}^k - \mathbf{w}^*|| + ||\mathbf{d}_w||)$;
(iii) *The sequence* $\{B_k\}$ *satisfies conditions* B.1 *and* B.2.
Then there is an $\varepsilon > 0$ *such that if* $||\mathbf{w}^0 - \mathbf{w}^*|| < \varepsilon$ *and* $||B_0 - \mathscr{H}L_*|| < \varepsilon$ *then the sequence* $\{\mathbf{w}^k\}$ *converges to* $\mathbf{w}^*$ *and* $\{(\boldsymbol{x}^k, \boldsymbol{z}^k)\}$ *converges Q-linearly to* $(\boldsymbol{x}^*, \boldsymbol{z}^*)$.

As linear convergence can be so slow as to be unsatisfactory, it is usually desirable to identify conditions under which a faster rate of convergence, namely superlinear convergence, is theoretically possible. Although in practical terms these conditions may not be achievable, they do suggest procedures that can lead to fast linear convergence. The following theorem gives a characterization of superlinear convergence. We use the notation $\mathscr{P}_k$ to denote the projection matrices that project vectors from $\mathscr{R}^n$ onto the null space of $D(\boldsymbol{x}^k)^{\mathrm{t}}$.

Theorem 4.2. *Suppose that the hypotheses of Theorem* 4.1 *hold. Then the convergence in* $\{(\boldsymbol{x}^k, \boldsymbol{z}^k)\}$ *is Q-superlinear if and only if*

$$\mathscr{P}_k(B_k - \mathscr{H}L_k)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = \mathbf{o}(||\boldsymbol{x}^{k+1} - \boldsymbol{x}^k|| + ||\boldsymbol{z}^{k+1} - \boldsymbol{z}^k||). \tag{16}$$

These theorems are a composite of the results of several authors ([6,5,16,9]). As the convergence results depend on conditions that may or may not hold for a particular SQP algorithm we comment on these restrictions separately.

Condition (i) of Theorem 4.1 requires that the step lengths go to unity as the solution is approached. Step lengths less than one may be required in these algorithms for two reasons. First, as is seen in the next section, global convergence considerations generally dictate that the step length be restricted so that some type of merit function be decreased at each step. Typically, the merit function depends only on $\boldsymbol{x}$ and $\boldsymbol{z}$ so that the restriction is on the size of the step $(\boldsymbol{d}_x, \boldsymbol{d}_z)$. Second, the multiplier and slack variables are usually maintained as positive (or at least nonnegative) throughout the iterations so that they will be nonnegative at optimality (which is required for feasibility and the identification of a minimum point). These restrictions put limits on the step length for the steps $\boldsymbol{d}_z$ and $\boldsymbol{d}_v$. The proof that the step lengths approach unity is very much dependent on the particular algorithm under consideration ([3,16]).

The satisfaction of condition (ii) also depends upon the particular form of the algorithm, specifically on the choice of approximation to the complementary slackness condition, i.e., (11), (12) or (14). It is easy to see that, under our assumptions, (11) satisfies (ii). On the other hand, whether the other approximations do clearly depends on the choice of ρ (as discussed in [16].) In addition, if, as is usually the case in large-scale problems, the quadratic problem is not solved exactly then the convergence analysis can be carried out only if the accuracy to which the solution is computed can be expressed in the form of $\mathbf{p}$.

Finally, the way in which the matrix B_k approximates $\mathscr{H}L_k$ also affects the convergence rate of the iterates. There has been a large amount of research on this issue (see [5] for an earlier review of the

literature), but the results are not totally satisfactory. Because it is much easier to solve the quadratic subproblem if the matrix B_k is positive definite, efforts have been made to generalize, for example, the BFGS and DFP updates of unconstrained optimization to the constrained case. These satisfy conditions B.1 and B.2 and work well in the case where the problem (NLP) is convex, but do not satisfy (16) in all cases. A constrained version of the PBS update has been shown to satisfy these two conditions and thus yield superlinear convergence, but has not been considered a satisfactory update owing to the nonconvexity of the quadratic approximation and its poor performance in practice.

5. Global convergence

An algorithm is said to be "globally convergent" if it converges from an arbitrary initial point to a local minimum. The procedures discussed in the preceding section will generally fail if the initial estimate is not near the solution because Newton's method is only locally convergent. To obtain global convergence, it is necessary to have some means of forcing a prospective new iterate, $\boldsymbol{x}^{k+1}$, to be a better approximation to $\boldsymbol{x}^*$ than is $\boldsymbol{x}^k$. The standard way of doing this is through the use of a *merit function*. The majority of this section will be devoted to an analysis of merit functions and their properties; at the end we will briefly discuss the idea of a "nonlinear filter", a recent development that provides a radically different approach to obtaining global convergence. It is important to point out that to make an SQP method effective, any procedures implemented to force global convergence should not impede the local convergence rate as the solution is neared.

A merit function is an auxiliary, scalar-valued function, $\phi(\boldsymbol{x})$, that has the property that if $\phi(\boldsymbol{x}^{k+1}) < \phi(\boldsymbol{x}^k)$, then $\boldsymbol{x}^{k+1}$ is acceptable as the next iterate. To ensure that reduction in ϕ implies progress, one constructs ϕ in such a way that the unconstrained minimizers of ϕ correspond to local solutions of (NLP) and the step $\boldsymbol{d}_x$ generated by the SQP method is a descent direction for ϕ. The natural merit function in unconstrained minimization is the function itself. In constrained optimization, the merit function must blend the need to reduce the objective function with the need to satisfy the constraints. Below we consider the properties of some of the more common examples of merit functions. To simplify the presentation, we first consider equality-constrained problems.

One of the earliest proposed merit functions is the ℓ_1 penalty function given by

$$\phi_1(\boldsymbol{x};\eta) = f(\boldsymbol{x}) + \eta||\boldsymbol{h}(\boldsymbol{x})||_1,$$

where η is a scalar to be chosen. For a point $\boldsymbol{x}^k$ such that $\boldsymbol{h}(\boldsymbol{x}^k) \neq 0$ and η sufficiently large, reducing ϕ_1 implies that $||\boldsymbol{h}(\boldsymbol{x}^k)||_1$ must be reduced. It can be shown that for η sufficiently large an unconstrained minimizer of this function corresponds to a solution of (NLP). This merit function has the disadvantage that it is not differentiable at feasible points.

Smoother merit functions, based on the augmented Lagrangian functions, offer several advantages that have caused them to be extensively studied. We illustrate the class with a simple version given by

$$\phi_F(\boldsymbol{x};\eta) = f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^t\bar{u}(\boldsymbol{x}) + \tfrac{1}{2}\eta||\boldsymbol{h}(\boldsymbol{x})||_2^2,$$

where, again, η is a constant to be specified and $\bar{\boldsymbol{u}}(\boldsymbol{x}) = -[\nabla\boldsymbol{h}(\boldsymbol{x})^t\nabla\boldsymbol{h}(\boldsymbol{x})]^{-1}\nabla\boldsymbol{h}(\boldsymbol{x})\nabla f(\boldsymbol{x})$. Observe that $\bar{\boldsymbol{u}}(\boldsymbol{x})$ is the least-squares estimate of the multipliers based on the first-order conditions. Thus the

first two terms of ϕ_F can be regarded as the Lagrangian and the last term augments the Lagrangian with a penalty term that is zero when the constraints are satisfied.

To show global convergence, we must first make some additional assumptions on the problem. These assumptions allow us to focus on the algorithms and not on the problem structure. They are:

C.1 All iterates $\boldsymbol{x}^k$ lie in a compact set $\mathscr{C}$.
C.2 The columns of $\nabla \boldsymbol{h}(\boldsymbol{x})$ are linearly independent for all $\boldsymbol{x} \in \mathscr{C}$.

The first assumption simply eliminates the possibility of a sequence of iterates diverging to infinity. This assumption, or something that implies it, is common in almost all global convergence analyses. The second assumption ensures that the linearized constraints are consistent, i.e., that the quadratic programming subproblems can be solved. Further comments on this matter are in Section 6.

Clearly the simple reduction of the merit function is not sufficient to obtain convergence, since then it would be possible for the procedure to stall at a nonoptimal point where α goes to zero. There are a number of conditions (such as the Armijo–Goldstein or Wolfe conditions) that can be imposed to ensure sufficient decrease in the merit function and to keep α bounded away from zero. A relatively simple set of conditions that might be used are given here. Suppose that for a given scalar-valued function $\phi(\boldsymbol{x})$, the sequence $\{\boldsymbol{x}^k\}$ is generated by $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}_{\boldsymbol{x}}^k$ where $\boldsymbol{d}_{\boldsymbol{x}}^k$ is a descent direction for ϕ at $\boldsymbol{x}^k$ and is $O||(\nabla\phi(\boldsymbol{x}^k))||$. Then, if for fixed $\sigma \in (0,1)$, the α_k are chosen by a backtracking line search to satisfy

$$\phi(\boldsymbol{x}^{k+1}) \leqslant \phi(\boldsymbol{x}^k) + \alpha\sigma_1 \nabla\phi(\boldsymbol{x}^k)^t \boldsymbol{d}_x, \tag{17}$$

it follows that

$$\lim_{k\to\infty} \nabla\phi(\boldsymbol{x}^k) = 0.$$

Thus any limit point of the sequence $\{\boldsymbol{x}^k\}$ is a critical point of ϕ.

It is possible that such a critical point will not be a local minimizer. Since the merit function is being reduced, this situation is rare, but precautions to ensure that a minimum has been achieved can be taken. See [18] for a more complete discussion.

In its simplest form the basic SQP algorithm that uses a merit function can be stated as follows: Solve the quadratic programming approximation for the step $\boldsymbol{d}_{\boldsymbol{x}}$; choose α bounded away form 0 to satisfy (17); repeat. Given the above assumptions we can state the following results for the merit function ϕ_F and this prototypical SQP algorithm:

(i) $\boldsymbol{x}^* \in \mathscr{C}$ is a local minimum of ϕ_F if and only if $\boldsymbol{x}^*$ is a local minimum of (NLP).
(ii) If $\boldsymbol{x}$ is not a critical point of (NLP), then $\boldsymbol{d}_{\boldsymbol{x}}$ is a descent direction for ϕ_F.
(iii) For η sufficiently large condition (17) can be satisfied for α bounded away from zero.
(iv) For η sufficiently large, the basic SQP algorithm is globally convergent to a critical point of (NLP).

Similar results hold for the merit function ϕ_1, but are slightly more difficult to state due to the nondifferentiability. In fact, such results also hold for the ℓ_p penalty function $\phi_p(\boldsymbol{x}) = f(\boldsymbol{x}) + \eta||\boldsymbol{h}(\boldsymbol{x})||_p$, where $p > 0$. (See [14] for a study of the ℓ_2 merit function.)

As stated earlier, the global convergence procedures should not conflict with the local convergence of Section 4. To achieve superlinear convergence requires, at the least, that the step lengths approach one as a solution is neared and that the Hessian approximations B_k satisfy the condition of Theorem

4.2. This is a difficult issue to resolve since it depends upon the merit function being used, the Hessian approximation, and the conditions for the acceptance of the step (e.g., (17)). For the nondifferentiable merit functions, a step length of one may not be acceptable no matter how close to the solution and no matter how good the Hessian approximation. This is called the Maratos effect [8]. Generally, it is not possible to prove superlinear convergence for a given SQP algorithm; one is forced to prove the weaker result that the merit function will allow a step length of one if superlinear convergence is possible, i.e., if the Hessian approximations satisfy (16). See [3] for an example of such an analysis.

In creating an efficient implementation of an SQP algorithm, it is necessary to make many decisions on the details. Several examples dealing with merit functions illustrate this point. First, a difficulty with the merit functions described above is that they involve a parameter, η, that must be adjusted as the iterations proceed. If the parameter is too large, there is no problem with the theory, but in practice, progress may be substantially slowed. If it is too small, then the merit function may not be adequate. Since the proper size may change from a remote starting guess to the solution, most successful implementations have heuristic adjustment procedures. These procedures usually perform well, but often lack theoretical justification [10,4]. As a second example, it has long been observed in nonlinear optimization that enforcing strict decrease in the merit function can sometimes lead to slow convergence and that allowing some occasional increases could improve the overall performance and even overcome the Maratos effect. One might think that this would destroy the global convergence, but there are ways to implement such a *nonmonotone* strategy that preserve global convergence, for instance, by insisting on sufficient decrease only after every K steps. Finally, some merit functions, including ϕ_{F}, are expensive to evaluate due to the gradient terms. In such cases, one may consider an approximate merit function at each iteration that is cheaper to evaluate, but sufficient to obtain global convergence [3,4,10].

When there are inequality constraints, constructing a merit function is more complicated. Theoretically, the correct active set will be identified by the quadratic program when the iterates are close enough to the solution. In problems with a large number of inequality constraints, however, it can often take many iterations to determine the correct active set. Thus inequality-constrained problems are, in this sense, harder than those with only equality constraints. Mathematically, inequalities can be partially eliminated by the nonnegative slack-variable techniques used in Section 3. A merit function can be constructed for inequality-constrained problems by using the quantity $||\boldsymbol{g}(\boldsymbol{x})_+||$ where the ith component of $\boldsymbol{g}(\boldsymbol{x})_+$ is 0 if $\boldsymbol{g}_i(\boldsymbol{x}) \leqslant 0$. This leads to merit functions that are not differentiable. (See [5] for further discussion.)

For SQP algorithms that use the trust-region approach, the step length parameter is not used. In this case the merit function is used in the determination of the trust-region radius, τ, given in (7). In unconstrained optimization, τ represents the radius of a ball about the current iterate in which a quadratic approximation is "trusted" to reflect (NLP) accurately. Similarly, in the constrained case, the trust-region parameter is modified at each step based on the accuracy with which the "predicted" decrease in the merit function fits the actual decrease in the merit function. For details see [7].

As noted above, virtually all merit functions involve a parameter that must be adaptively chosen. Recently the idea of using a "nonlinear filter", in a trust-region method has been suggested as an alternative to a merit function for a problem with inequality constraints. For such a problem the pair (r_k, f_k) is computed at each iterate $\boldsymbol{x}^k$ where

$$r(\boldsymbol{g}(\boldsymbol{x})) = \max\{0, \max\{\boldsymbol{g}_i(\boldsymbol{x}),\ i = 1, 2, \ldots, m\}\} \tag{18}$$

defines a measure of infeasibility at the vector $\boldsymbol{x}$. A pair (r_i, f_i) is said to *dominate* the pair (r_j, f_j) if and only if $r_i \leqslant r_j$ and $f_i \leqslant f_j$. This indicates that the pair (r_i, f_i) is at least as good as (r_j, f_j) in that the objective function value is at least as small and the constraint violations are no larger. A *filter* is a list of pairs (r_i, f_i) such that no point in the list dominates any other point. As the algorithm proceeds, a pair (r_k, f_k) is added to the filter if its corresponding filter pair is not dominated by any pair in the filter. If it is added to the filter, all pairs in the filter that it dominates are removed. If the point is not acceptable, the trust-region radius is reduced. The advantage of such a technique is that it does not require the selection and adjustment of any penalty parameter. More details of using this idea can be found in [11].

6. Solvers for quadratic programs

A key aspect of an SQP algorithm is the quadratic program (QP) solver, or equivalently, the solvers for any of the formulations in Section 3. For a QP solver to be effective in the large-scale case, it should have several desirable properties. First, it must be a computationally efficient method and it should be tailored to the specific type of quadratic program arising from (NLP). For large problems, the ability to solve the QPs approximately may lead to substantial improvements in the overall efficiency. Thus there should be criteria that allow the solver to halt early. This, in turn, requires that the SQP method be coordinated with the QP solver in the sense that the approximate solution must still be a descent direction for the merit function or be a useful direction for the SQP algorithm. All QP solvers must detect inconsistent constraints and should take some action to generate a useful step. (Note that trust-region methods in the constrained case can readily cause inconsistencies when the point $\boldsymbol{x}_k$ is not feasible and τ is small.) Remedial action is often accomplished by perturbing the constraints in some way and solving the perturbed problem. Finally, for very large problems, the QP solver should be able to exploit parallelism. The current state of the art suggests that active set or simplex-based methods are not readily parallelizable, whereas interior-point methods lend themselves to parallel environments.

If the final active set for a quadratic program were known, then the solution could be found by solving a single system of linear equations. Thus a standard approach to solving quadratic programs is to use an "active set" method that works from an estimate of the final active set, called the *working set*. The quadratic program is solved assuming that these are equality constraints, ignoring the rest. At this solution, new constraints encountered are added to the working set and some of the current constraints are dropped, depending on the sign of the multipliers. A factorization of the matrix of constraint gradients associated with the working set is usually required [12], but iterative methods can also be used for the linear systems [13]. An advantage of these methods is that the active set from the previous iteration of the SQP algorithm is often a good estimate of the active set at the current iteration. As noted earlier, the active set for (NLP) is identified as the solution is neared, so active set methods tend to be extremely efficient over the final few iterations.

Recently there have been successful implementations of both primal and primal-dual interior-point methods for QPs. (Primal-dual interior-point methods are based on (NLP) (ρ) given in Section 2.) For these methods difficulties can arise in the nonconvex case [19,20]. There is also a purely primal method that works for both convex and nonconvex problems. This method solves a QP by solving a sequence of three-dimensional approximations to the QP [2].

7. Reduced Hessian formulation and applications

In some applications, the optimization problem is an equality-constrained problem characterized by having a large number of equality constraints, m, relative to the number of variables, n, i.e., $n-m$ is relatively small. In this case, the quadratic program (QP1) becomes

$$\begin{aligned} &\underset{\boldsymbol{d}_x}{\text{minimize}} \quad \nabla f(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \tfrac{1}{2}\boldsymbol{d}_x^{\mathrm{t}} B_k \boldsymbol{d}_x \\ &\text{subject to } \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \boldsymbol{h}(\boldsymbol{x}^k) = 0. \end{aligned} \tag{19}$$

Assuming that $\nabla \boldsymbol{h}(\boldsymbol{x}^k)$ has full column rank, let the columns of $Z \in \Re^{n\times(n-m)}$ be a basis for the null space of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}$ and let the columns of $Y \in \Re^{n\times m}$ be a basis of the range space of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)$. Decomposing the vector $\boldsymbol{d}_x$ as

$$\boldsymbol{d}_x = Z\boldsymbol{r}_Z + Y\boldsymbol{r}_Y,$$

for vectors $\boldsymbol{r}_Z \in \Re^{n-m}$ and $\boldsymbol{r}_Y \in \Re^m$, the constraint equation in (19) can be written as

$$\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} Y \boldsymbol{r}_Y + \boldsymbol{h}(\boldsymbol{x}^k) = 0,$$

which can be solved to obtain

$$\boldsymbol{r}_Y = -[\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} Y]^{-1}\boldsymbol{h}(\boldsymbol{x}^k).$$

Thus (19) becomes an unconstrained minimization problem in the $(n-m)$ variables $\boldsymbol{r}_Z$ given by

$$\underset{\boldsymbol{r}_Z}{\text{minimize}} \quad \tfrac{1}{2}\boldsymbol{r}_Z^{\mathrm{t}}[Z^{\mathrm{t}}B_k Z]\boldsymbol{r}_Z + (\nabla f(\boldsymbol{x}^k) + B_k Y \boldsymbol{r}_Y)^{\mathrm{t}} Z \boldsymbol{r}_Z.$$

The $(n-m)\times(n-m)$ matrix $Z^{\mathrm{t}}B_k Z$ is called the *reduced Hessian* and the $(n-m)$ vector $Z^{\mathrm{t}}(\nabla f(\boldsymbol{x}^k) + B_k Y \boldsymbol{r}_Y)$ is called the *reduced gradient.* The advantage of this formulation is that the reduced Hessian, under our assumptions, is positive definite at the solution. (It is not unique, however, since it depends on the choice of the basis for the null space, Z.) It thus makes sense to approximate the reduced Hessian by updates that maintain positive definiteness. The specific details can vary; see, e.g., [1,12–14].

An application of this idea occurs when solving optimization problems where the objective function and/or the constraint functions require the solution of a partial differential equation (PDE). The function to be optimized depends both on a set of control or design parameters and on a set of state variables. These sets of variables are related through a PDE. To be more specific, let $f(\boldsymbol{x}, \boldsymbol{c})$ be the function to be optimized. Here the vector $\boldsymbol{c} \in \Re^q$ represents the design or control variables and $\boldsymbol{x} \in \Re^n$ represents the state variables. The state variables satisfy a differential equation that is represented as a discretized operator yielding $S(\boldsymbol{x}, \boldsymbol{c}) = 0$, where $S : \Re^{n+q} \to \Re^n$. In this context, it can be assumed that for $\boldsymbol{c}$ restricted to a given set, this equation can be solved for a unique $\boldsymbol{x}(\boldsymbol{c})$. Thus the resulting optimization problem is

$$\begin{aligned} &\underset{(\boldsymbol{x},\boldsymbol{c})}{\text{minimize}} \quad f(\boldsymbol{x}, \boldsymbol{c}) \\ &\text{subject to } S(\boldsymbol{x}, \boldsymbol{c}) = 0, \end{aligned} \tag{20}$$

where there are $(n+q)$ variables and n constraints. Typically q is small compared to n.

The particular structure of the equality constraints allows for a variety of possible versions of the reduced-Hessian SQP algorithm [21]. Similar problem forms arise in related applications including

parameter identification and inverse problems. In such applications, there are also some inequality constraints that bound the allowable range of the c components or that restrict some other function of the variables. In some cases, the PDE problems cannot be solved if these constraints are violated and it is necessary to remain feasible with respect to these constraints. A special version of SQP, called FSQP for feasible SQP, is designed to maintain feasibility [15].

References

[1] L.T. Biegler, J. Nocedal, C. Schmid, A reduced Hessian method for large-scale constrained optimization, SIAM J. Optim. 5 (2) (1995) 314–347.

[2] P.T. Boggs, P.D. Domich, J.E. Rogers, An interior-point method for general large scale quadratic programming problems, Ann. Oper. Res. 62 (1996) 419–437.

[3] P.T. Boggs, A.J. Kearsley, J.W. Tolle, A global convergence analysis of an algorithm for large scale nonlinear programming problems, SIAM. J. Optim. 9 (4) (1999) 833–862.

[4] P.T. Boggs. A.J. Kearsley, J.W. Tolle, A practical algorithm for general large scale nonlinear optimization problems, SIAM J. Optim. 9 (3) (1999) 755–778.

[5] P.T. Boggs, J.W. Tolle, Sequential quadratic programming, Acta Numer. 1995 (1995) 1–52.

[6] P.T. Boggs, J.W. Tolle, P. Wang, On the local convergence of quasi-Newton methods for constrained optimizations, SIAM J. Control Optim. 20 (1982) 161–171.

[7] R.H. Byrd, J.C. Gilbert, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, Technical Report OTC 96-02, Northwestern University, August 1998.

[8] R. Chamberlain, C. Lemarechal, H.C. Pedersen, M.J.D. Powell, The watchdog technique for forcing convergence in algorithms for constrained optimization, Math. Programming Study 16 (1982) 1–17.

[9] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM J. Numer. Anal. 19 (1982) 400–408.

[10] M. El-Alem, A robust trust-region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization, SIAM J. Optim. 5 (2) (1995) 348–378.

[11] R. Fletcher, S. Leyffer, P. Toint, On the global convergence of an SLP-filter algorithm, Report, University of Dundee, 1998.

[12] P. Gill, W. Murray, M. Saunders, SNOPT: an SQP algorithm for large scale constrained optimization, preprint NA97-2, University of California, San Diego, 1997.

[13] N.I.M. Gould, M.E. Hribar, J. Nocedal, On the solution of equality constrained quadratic programming problems arising in optimization, Technical Report OTC 98/06, Argonne National Laboratory and Northwestern University, 1998.

[14] M. Lalee, J. Nocedal, T. Plantenga, On the implementation of an algorithm for large-scale equality constrained optimization, SIAM J. Optim. 8 (3) (1998) 682–706.

[15] C.T. Lawrence, A.T. Tits, A computationally efficient feasible sequential quadratic programming algorithm, preprint, University of Maryland, 1998.

[16] H.J. Martinez, Z. Parada, R.A. Tapia, On the characterization of q-superlinear convergence of quasi-Newton interior-point methods for nonlinear programming, Bol. Soc. Mat. Mexicana 1 (3) (1995) 137–148.

[17] S.G. Nash, A. Sofer, Linear and Nonlinear Programming, McGraw-Hill, New York, 1995.

[18] J. Nocedal, Theory of algorithms for unconstrained optimization, Acta Numer. 1991 (1992) 199–242.

[19] D.F. Shanno, R.J. Vanderbei, An interior-point algorithm for nonconvex nonlinear programming, preprint, Princeton University, 1998.

[20] R.J. Vanderbei, LOQO: an interior-point code for quadratic programming, Technical Report SOR 94-15, Princeton University, 1994.

[21] D.P. Young, D.E. Keyes, Newton's method and design optimization, preprint, Old Dominion University, 1998.