IMA Journal of Numerical Analysis (2012) **32**, 553–579
doi:10.1093/imanum/drq037
Advance Access publication on August 20, 2011

A sequential quadratic programming algorithm with an additional equality constrained phase

JOSÉ LUIS MORALES
Departamento de Matemáticas, Instituto Tecnológico Autónomo de México, Mexico City 01080, México

AND

JORGE NOCEDAL* AND YUCHEN WU
Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208-3118, USA
*Corresponding author: nocedal@eecs.northwestern.edu

[Received on 6 February 2010; revised on 2 July 2010]

A sequential quadratic programming (SQP) method is presented that aims to overcome some of the drawbacks of contemporary SQP methods. It avoids the difficulties associated with indefinite quadratic programming subproblems by defining this subproblem to be always convex. The novel feature of the approach is the addition of an equality constrained quadratic programming (EQP) phase that promotes fast convergence and improves performance in the presence of ill conditioning. This EQP phase uses exact second-order information and can be implemented using either a direct solve or an iterative method. The paper studies the global and local convergence properties of the new algorithm and presents a set of numerical experiments to illustrate its practical performance.

Keywords: constrained optimization; nonlinear programming; sequential quadratic programming.

1. Introduction

Sequential quadratic programming (SQP) methods are very effective techniques for solving small-size, medium-size and certain classes of large-scale nonlinear programming problems. They are often preferable to interior point methods when a sequence of related problems must be solved (as in branch and bound methods) and more generally, when a good estimate of the solution is available. Some SQP methods employ convex quadratic programming subproblems for the step computation (typically using quasi-Newton Hessian approximations), while other variants define the Hessian of the SQP model using second derivative information, which can lead to nonconvex quadratic subproblems; see Gould *et al.* (2005) and Boggs & Tolle (1995) for surveys on SQP methods.

All these approaches have drawbacks. SQP methods that employ convex quasi-Newton approximations (Schittkowski, 1981; Drud, 1985; Gill *et al.*, 2002) can be slow when solving large or badly scaled problems, whereas methods that employ the exact Hessian of the Lagrangian (Fletcher & Leyffer, 2002) are confronted with the difficult task of computing solutions of indefinite quadratic programs (Gould *et al.*, 2005). In this paper we propose an algorithm that aims to overcome some of these drawbacks by formulating the SQP iteration in two phases. The algorithm first solves a convex quadratic program to estimate the optimal active set and then employs second derivative information in an additional equality

constrained phase (EQP) that promotes rapid convergence. The EQP phase is normally less expensive to solve than a general quadratic program and allows for the application of iterative methods that scale up well in the number of variables. We call our algorithm SQP+ because of the incorporation of the additional EQP phase.

Concurrently with this work, Gould & Robinson (2010a,b) have developed a class of two-phase SQP methods for large-scale optimization. Their main focus is on trust region algorithms that solve two inequality constrained quadratic programs at every iteration. In their approach a convex quadratic program is first solved to yield a so-called predictor step, which is then used to guide the solution of a second quadratic program (constructed using second derivative information). Gould and Robinson have also proposed a method similar to SQP+ in which the second phase of the algorithm consists of an EQP step. Other SQP methods that perform the step computation in two phases are discussed in Facchinei & Lucidi (1994) and Friedlander *et al.* (2007).

The nonlinear programming problem under consideration is stated as

$$\begin{aligned} &\min_x \ f(x) \\ &\text{subject to } \ h(x) = 0, \\ &\qquad\quad g(x) \geqslant 0, \end{aligned} \tag{1.1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$, $h\colon \mathbb{R}^n \to \mathbb{R}^m$ and $g\colon \mathbb{R}^n \to \mathbb{R}^t$ are smooth functions. We write the Lagrangian of this problem as

$$L(x, \lambda, \mu) = f(x) - \lambda^{\mathrm{T}} h(x) - \mu^{\mathrm{T}} g(x), \tag{1.2}$$

where λ and μ are vectors of multipliers. The Karush-Kuhn-Tucker (KKT) conditions for the nonlinear program (1.1) are given by

$$\nabla f(x) - H(x)^{\mathrm{T}}\lambda - G(x)^{\mathrm{T}}\mu = 0, \tag{1.3a}$$

$$h(x) = 0, \tag{1.3b}$$

$$g(x) \geqslant 0, \tag{1.3c}$$

$$\mu \geqslant 0, \tag{1.3d}$$

$$\mu^{\mathrm{T}} g(x) = 0, \tag{1.3e}$$

where H and G are the Jacobian matrices of h and g, respectively.

The SQP approach presented in this paper can be implemented both in a line search or in a trust region framework. In either case global convergence can be promoted by imposing decrease in the ℓ_1 merit function

$$\phi_\pi(x) = f(x) + \pi \|h(x)\|_1 + \pi \|g(x)^-\|_1, \tag{1.4}$$

where $g(x)^- \triangleq \max\{0, \ -g_i(x)\}$ and $\pi > 0$ is a penalty parameter.

The paper is organized in six sections. In Section 2 we describe the proposed SQP method and outline various implementation options. In Sections 3 and 4 we discuss the global and local convergence properties of the new algorithm, and in Section 5 we report the results of some numerical experiments. Some concluding remarks are made in Section 6.

2. The SQP+ method

The SQP+ approach can be implemented in a line search or trust region framework. For concreteness, we consider only a line search implementation in this paper. Given a primal–dual iterate (x_k, λ_k, μ_k), the algorithm first computes a step d_k^{IQ} in the primal variables x by solving the standard quadratic programming subproblem

$$\min_d \quad \nabla f(x_k)^{\mathrm{T}} d + d^{\mathrm{T}} B_k d \tag{2.1a}$$

$$\text{subject to} \quad h(x_k) + H(x_k) d = 0, \tag{2.1b}$$

$$g(x_k) + G(x_k) d \geqslant 0, \tag{2.1c}$$

where B_k is a *positive definite* approximation to the Hessian of the Lagrangian $\nabla^2_{xx} L(x_k, \lambda_k, \mu_k)$. The multipliers associated with the solution of (2.1) are denoted by $\lambda_{k+1}^{\text{IQ}}$, μ_{k+1}^{IQ} and will be referred to as the 'IQP multipliers' (Gill *et al.*, 1981). We assume that the constraints (2.1b)–(2.1c) are feasible and will later discuss how to handle the case when they are not. One of the goals of this inequality constrained quadratic programming (IQP) phase is to provide a good estimate, say $\mathcal{W}_k$, of the optimal active set. To this end we solve problem (2.1) accurately and define $\mathcal{W}_k$ as the indices of the constraints (2.1b)–(2.1c) that are satisfied as equalities at the solution d_k^{IQ}, i.e.,

$$\mathcal{W}_k = \{i \in \mathcal{E}\} \cup \left\{i \in \mathcal{I} | g_i(x_k) + \nabla g_i(x_k)^{\mathrm{T}} d_k^{\text{IQ}} = 0\right\}, \tag{2.2}$$

where $\mathcal{E}$ and $\mathcal{I}$ are the index sets of the equality and inequality constraints, respectively. To compensate for the limitations of the IQP phase and to promote a fast rate of convergence, the algorithm computes an additional step by solving an equality constrained quadratic problem (EQP) in which the constraints in $\mathcal{W}_k$ are imposed as equalities and all other constraints are (temporarily) ignored. The EQP subproblem is given by

$$\min_d \quad \left(\nabla f(x_k) + W_k d_k^{\text{IQ}}\right)^{\mathrm{T}} d + \tfrac{1}{2} d^{\mathrm{T}} W_k d \tag{2.3a}$$

$$\text{subject to } \nabla h_i(x_k)^{\mathrm{T}} d = 0, \quad i \in \mathcal{E}, \tag{2.3b}$$

$$\nabla g_i(x_k)^{\mathrm{T}} d = 0, \quad i \in \mathcal{W}_k \cap \mathcal{I}, \tag{2.3c}$$

where

$$W_k = W\left(x_k, \lambda_{k+1}^{\text{IQ}}, \mu_{k+1}^{\text{IQ}}\right) \quad \text{with} \quad W(x, \lambda, \mu) \triangleq \nabla^2_{xx} L(x, \lambda, \mu), \tag{2.4}$$

and $\nabla h_i(x_k)^{\mathrm{T}}$ and $\nabla g_i(x_k)^{\mathrm{T}}$ denote the rows of the matrices $H(x_k)$ and $G(x_k)$, respectively. Problem (2.3) could be unbounded because, away from the solution, the Hessian of the Lagrangian may not be positive definite on the tangent space of the constraints defined by $\mathcal{W}_k$. In this case we can add a regularization term to the objective (2.3a) or include a trust region constraint in problem (2.3) to ensure that the EQP subproblem is well defined.

We denote an approximate solution to (2.3) by d_k^{EQ} and the corresponding Lagrange multipliers by $\lambda_{k+1}^{\text{EQ}}$, μ_{k+1}^{EQ}. In Section 3 we show that there is much freedom in the selection of the EQP step. There are two effective procedures for computing d_k^{EQ}, namely a projected conjugate gradient method (Keller *et al.*, 2000; Coleman & Verma, 2001; Gould *et al.*, 2001) and the direct solution of the (possibly modified) KKT system of (2.3) (Bonnans & Launay, 1995; Vanderbei & Shanno, 1999).

Next the algorithm computes a contraction parameter $\beta \in [0, 1]$ such that the step $d = d_k^{\mathrm{IQ}} + \beta d_k^{\mathrm{EQ}}$ satisfies all linearized constraints that are not in the working set $\mathcal{W}_k$, i.e.,

$$g_i(x_k) + \nabla g_i(x_k)^{\mathrm{T}} d \geqslant 0, \quad i \in \mathcal{W}_k^{\mathrm{c}}, \tag{2.5}$$

where $\mathcal{W}_k^{\mathrm{c}}$ denotes the complement of $\mathcal{W}_k$. Thus, all the steps of the algorithm will satisfy the linearized constraints (2.1b)–(2.1c).

Before the line search is performed, the algorithm updates the penalty parameter π in (1.4). For this purpose we define a model of the merit function ϕ_π around the point x_k as

$$q_\pi(d) = f_k + \nabla f_k^{\mathrm{T}} d + \tfrac{1}{2} d^{\mathrm{T}} B_k d + \pi \|h_k + H_k d\|_1 + \pi \|[g_k + G_k d]^-\|_1, \tag{2.6}$$

where f_k is shorthand for $f(x_k)$ and similarly for other functions. We define the model reduction from x_k to $x_k + d$, by

$$qred(d) = q_\pi(0) - q_\pi(d). \tag{2.7}$$

For a step that satisfies the linearized constraints (2.1b)–(2.1c), we have

$$qred(d) = -\nabla f_k^{\mathrm{T}} d - \tfrac{1}{2} d^{\mathrm{T}} B_k d + \pi \left(\|h_k\|_1 + \left\|g_k^-\right\|_1\right). \tag{2.8}$$

The update of the penalty parameter is based on the IQP step. As is now common (Burke & Han, 1989; Waltz *et al.*, 2006; Byrd *et al.*, 2008), we require that the new penalty parameter π_k be large enough that

$$qred(d_k^{\mathrm{IQ}}) \geqslant \rho \pi_k \left(\|h_k\|_1 + \left\|g_k^-\right\|_1\right), \tag{2.9}$$

for some prescribed constant $\rho \in (0, 1)$. From (2.8) we see that condition (2.9) is equivalent to the requirement

$$\pi_k \geqslant \frac{\nabla f_k^{\mathrm{T}} d_k^{\mathrm{IQ}} + \frac{1}{2}\left(d_k^{\mathrm{IQ}}\right)^{\mathrm{T}} B_k d_k^{\mathrm{IQ}}}{(1-\rho)\left(\|h_k\|_1 + \left\|g_k^-\right\|_1\right)} \triangleq \pi_{\mathrm{trial}}. \tag{2.10}$$

We can enforce this condition by updating the penalty parameter at every iteration k by means of the following rule:

$$\pi_k = \begin{cases} \pi_{k-1} & \text{if (2.9) is satisfied,} \\ \pi_{\mathrm{trial}} + \pi_b & \text{otherwise,} \end{cases} \tag{2.11}$$

where $\pi_b > 0$ is a given constant and π_{trial} is defined in (2.10).

The algorithm then performs a backtracking line search along the piecewise linear segment that starts at x_k, goes through $x_k + d_k^{\mathrm{IQ}}$ and ends at $x_k + d_k^{\mathrm{IQ}} + \beta d_k^{\mathrm{EQ}}$. The line search first attempts to find a steplength $\alpha_k \in [\alpha_{\min}, 1]$ that satisfies the sufficient decrease condition

$$\phi_{\pi_k}\left(x_k + d_k^{\mathrm{IQ}} + \alpha_k \beta d_k^{\mathrm{EQ}}\right) \leqslant \phi_{\pi_k}(x_k) - \sigma\, qred\left(d_k^{\mathrm{IQ}}\right), \tag{2.12}$$

for some given constant $\sigma \in (0, 1)$, and where $\alpha_{\min}$ is a threshold such as 0.25. If this line search is successful, we define the total step as

$$d_k = d_k^{\mathrm{IQ}} + \alpha_k \beta d_k^{\mathrm{EQ}}. \tag{2.13}$$

Otherwise, we choose a constant $\tau \in (0, 1)$ and let $\alpha_k \in (0, 1]$ be the first member of the sequence $\{1, \tau, \tau^2, \ldots\}$ such that

$$\phi_{\pi_k}\left(x_k + \alpha_k d_k^{\mathrm{IQ}}\right) \leqslant \phi_{\pi_k}(x_k) - \sigma \alpha_k qred\left(d_k^{\mathrm{IQ}}\right), \tag{2.14}$$

where σ is the same constant as in (2.12). We then set

$$d_k = \alpha_k d_k^{\mathrm{IQ}}. \tag{2.15}$$

Regardless of whether d_k is defined by (2.13) or (2.15), the new primal iterate is defined as

$$x_{k+1} = x_k + d_k. \tag{2.16}$$

Note that, unlike (2.14), the step length α_k does not appear in the sufficient decrease condition given by the last term in (2.12). This is because we want global convergence to be driven by the IQP phase and therefore accept the EQP step only if it gives as much reduction as that predicted by the full IQP step.

New Lagrange multipliers $(\lambda_{k+1}, \mu_{k+1})$ are defined at the EQP point as follows. First, we set the multipliers corresponding to the inactive linearized constraints to zero. The rest of the multipliers are set to the EQP multipliers $\lambda_{k+1}^{\mathrm{EQ}}$, μ_{k+1}^{EQ}—except that if any components of μ_{k+1}^{EQ} are negative, they are set to zero. The IQP multipliers are undeniably another reasonable candidate for the new Lagrange multipliers. But, we believe the EQP multipliers are better because the EQP subproblem (2.3) uses the exact Hessian of the Lagrangian W_k, yielding accurate curvature information. In addition, as we will show in Section 4, under suitable conditions the EQP multipliers converge rapidly to the optimal multipliers.

Before describing the algorithm in more detail we introduce some notation to denote subvectors of inequality constraints and their multipliers. Given a working set $\mathcal{W}_k$, and multipliers μ corresponding to the inequality constraints g in (1.3c) or (2.1c), we denote by $[g]_{\mathcal{W}_k \cap \mathcal{I}}$ the subvector of constraints g restricted to the set $\mathcal{W}_k \cap \mathcal{I}$ and by $[\mu]_{\mathcal{W}_k \cap \mathcal{I}}$ the corresponding multipliers. Similarly, we denote by $[g]_{\mathcal{W}_k^c}$ the subvector of constraints g restricted to the complement $\mathcal{W}_k^c$ and by $[\mu]_{\mathcal{W}_k^c}$ the corresponding multipliers. To further simplify notation we omit the square bracket '[ ]' and the subscript '$\cap\mathcal{I}$' when it is unambiguous to do so. Using this convention we have

$$g_{\mathcal{W}_k} \equiv [g]_{\mathcal{W}_k} \equiv [g]_{\mathcal{W}_k \cap \mathcal{I}} \text{ and } \mu_{\mathcal{W}_k} \equiv [\mu]_{\mathcal{W}_k} \equiv [\mu]_{\mathcal{W}_k \cap \mathcal{I}}.$$

The new SQP algorithm is specified as follows.

Algorithm I

Initial data: x_0, $\pi_0 > 0$, $\pi_b > 0$, $\rho > 0$, $\tau \in (0, 1)$ and $\sigma \in (0, 1)$.

For $k = 1, 2, \ldots$ until the KKT conditions (1.3) for the nonlinear program (1.1) are satisfied:

1. Define a positive definite matrix B_k and compute the step d_k^{IQ} and multipliers $\lambda_{k+1}^{\mathrm{IQ}}$, μ_{k+1}^{IQ} by solving the subproblem (2.1);
2. Determine the working set $\mathcal{W}_k$;
3. Compute the EQP step d_k^{EQ} and multipliers $\lambda_{k+1}^{\mathrm{EQ}}$, μ_{k+1}^{EQ} by finding an approximate solution of problem (2.3);
4. Compute the largest number $\beta \in [0, 1]$ that ensures that the step $d = d_k^{\mathrm{IQ}} + \beta d_k^{\mathrm{EQ}}$ satisfies the constraints (2.5);

5. Compute the penalty parameter π_k by (2.11);
6. Compute the step length α_k, define d_k by (2.13) or (2.15), and set $x_{k+1} = x_k + d_k$;
7. Set

$$\lambda_{k+1} = \lambda_{k+1}^{\text{EQ}}, \quad [\mu_{k+1}]_{\mathcal{W}_k} = \max(0, \mu_{k+1}^{\text{EQ}}), \quad [\mu_{k+1}]_{\mathcal{W}_k^c} = 0.$$

We have defined the model q_π in terms of the positive definite Hessian approximation B_k so that the IQP step drives the global convergence of the algorithm—while the EQP phase yields superlinear convergence.

Many enhancements and safeguards are needed to transform this general (and somewhat idealized) algorithm into a practical approach. For example, the constraints in (2.1) are not always feasible. This issue can be resolved by modifying the constraints using relaxations (Burke & Han, 1989; Omojokun, 1989) or by following a penalty approach (Fletcher, 1987; Chen & Goldfarb, 2006; Byrd *et al.*, 2009). The algorithm should also include regularization strategies for controlling singularity and ill conditioning (Gill *et al.*, 1981; Conn *et al.*, 2000), as well as second-order corrections (Fletcher, 1987) or watchdog steps (Chamberlain *et al.*, 1982) to improve the efficiency of the iteration. Step 4 is quite restrictive (for simplicity) and in a practical implementation we might allow the EQP step to violate some of the linearized constraints that are not in the working set. In Section 6 we comment on a variant that employs a projection in the EQP phase instead of backtracking to satisfy the constraints (2.5).

A fully developed implementation of the proposed approach is outside the scope of this paper. We focus, instead, on some of the key aspects of the algorithm so that its potential can be assessed in a simple setting. One of the main questions we wish to investigate is whether the EQP phase does, in fact, add significant benefits to the standard SQP approach.

One motivation for the SQP+ algorithm stems from the difficulties that have been encountered in trying to introduce second derivative information in SQP codes that were originally designed to solve convex quadratic subproblems, such as SNOPT. Another motivation arose from numerical experience with the sequential linear-quadratic programming (SLQP) method described in Byrd *et al.* (2004, 2006), which reveals the limitations of using only first-order information in the active set prediction phase. For example, it has proved to be difficult to warm start the linear programming phase in SLQP methods because the trust region constraint, which is defined as a box constraint (Fletcher & Sainz de la Maza, 1989; Chin & Fletcher, 2003; Byrd *et al.*, 2004), is typically active at every iteration and it is difficult to predict which sides of the box will be active. The IQP phase of the SQP+ algorithm is both easier to warm start and provides a better working set estimation than the SLQP method—often at the price of a marginally more expensive step computation.

3. Global convergence properties

In this section we show that Algorithm I enjoys favourable global convergence properties. We make the following assumptions about problem (1.1) and the iterates generated by the algorithm.

ASSUMPTION 3.1

(a) The sequence $\{x_k\}$ generated by Algorithm I is contained in a convex set Ω where the functions f, h, g are twice differentiable; the sequence $\{f_k\}$ is bounded below and the sequences $\{\nabla f_k\}$, $\{h_k, \}$, $\{g_k\}$, $\{H_k\}$ and $\{G_k\}$ are bounded.

(b) The quadratic program (2.1) is feasible for all k.

(c) The sequence of IQP multipliers is bounded, i.e., there exists a positive constant η such that $\|(\lambda_k^{\text{IQ}}, \mu_k^{\text{IQ}})\|_\infty \leqslant \eta$ for all k.

(d) The sequence of Hessian approximations $\{B_k\}$ is bounded above and there exists a constant $M > 0$ such that for all k,

$$\frac{1}{2}\left(d_k^{\text{IQ}}\right)^{\text{T}} B_k d_k^{\text{IQ}} \geqslant M \left\| d_k^{\text{IQ}} \right\|_2^2. \tag{3.1}$$

These assumptions are fairly restrictive, in particular (b) and (c). Although they have often been used in the literature (see, e.g., Powell, 1983), recent studies of SQP methods establish global convergence properties under weaker assumptions. This requires, however, that the SQP method be modified significantly by incorporating constraint relaxations, Hessian modifications, trust regions or other devices (Conn *et al.*, 2000; Chen & Goldfarb, 2006; Byrd *et al.*, 2009). The resulting algorithms are complex and the analysis is long and laborious. By making the above assumptions we can develop a fairly compact convergence analysis that highlights the key properties of the SQP+ approach.

We begin the analysis by noting that the primal–dual solution $(d_k^{\text{IQ}}, \lambda_{k+1}^{\text{IQ}}, \mu_{k+1}^{\text{IQ}})$ of the IQP subproblem (2.1) satisfies the following first-order conditions:

$$\nabla f_k + B_k d_k^{\text{IQ}} - {H_k}^{\text{T}} \lambda_{k+1}^{\text{IQ}} - {G_k}^{\text{T}} \mu_{k+1}^{\text{IQ}} = 0, \tag{3.2a}$$

$$h_k + H_k d_k^{\text{IQ}} = 0, \tag{3.2b}$$

$$g_k + G_k d_k^{\text{IQ}} \geqslant 0, \tag{3.2c}$$

$$\mu_{k+1}^{\text{IQ}} \geqslant 0, \tag{3.2d}$$

$$\left(\mu_{k+1}^{\text{IQ}}\right)^{\text{T}} \left(g_k + G_k d_k^{\text{IQ}}\right) = 0. \tag{3.2e}$$

It is not difficult to show (cf. Nocedal & Wright, 2006, p. 628) that the directional derivative of the merit function ϕ_{π_k} at a point x_k along the direction d_k^{IQ} satisfies

$$D\phi_{\pi_k}\left(x_k; d_k^{\text{IQ}}\right) \leqslant \nabla f_k^{\text{T}} d_k^{\text{IQ}} - \pi \|h(x_k)\|_1 - \pi \|g(x_k)^-\|_1.$$

By comparing the right-hand side of this expression with (2.8), we obtain

$$D\phi_{\pi_k}\left(x_k; d_k^{\text{IQ}}\right) \leqslant -qred\left(d_k^{\text{IQ}}\right) - \frac{1}{2} d_k^{\text{IQ}\text{T}} B_k d_k^{\text{IQ}}. \tag{3.3}$$

Recalling from (2.9) that $qred(d_k^{\text{IQ}}) \geqslant 0$ and since B_k is positive definite we conclude that d_k^{IQ} is a descent direction for ϕ_{π_k}. Relation (3.3) also shows that, by using $qred(d_k^{\text{IQ}})$ in the right-hand sides of (2.12) and (2.14) (instead of using the directional derivative $D\phi_{\pi_k}(x_k; d_k^{\text{IQ}})$ as in a standard Armijo condition), the sufficient decrease condition is less demanding and accounts for the nondifferentiability of the penalty function. This allows the algorithm to take longer steps.

An immediate consequence of Assumption 3.1(c) is that the penalty parameters π_k generated by the update rule (2.11) are bounded.

LEMMA 3.2 There is an index $\bar{k}$ and a positive constant $\bar{\pi}$ such that $\pi_k = \bar{\pi}$ for all $k \geqslant \bar{k}$.

Proof. From (3.2b) and (3.2e) we have that

$$\left(H_k d_k^{\text{IQ}}\right)^{\text{T}} \lambda_{k+1}^{\text{IQ}} = -h_k^{\text{T}} \lambda_{k+1}^{\text{IQ}}, \ \left(G_k d_k^{\text{IQ}}\right)^{\text{T}} \mu_{k+1}^{\text{IQ}} = -g_k^{\text{T}} \mu_{k+1}^{\text{IQ}}. \tag{3.4}$$

Next, since $\mu_{k+1}^{\text{IQ}} \geqslant 0$, we have that

$$-\left(\mu_{k+1}^{\text{IQ}}\right)^{\text{T}} g_k \leqslant \left(\mu_{k+1}^{\text{IQ}}\right)^{\text{T}} \max\{0, -g_k\} = \left(\mu_{k+1}^{\text{IQ}}\right)^{\text{T}} g_k^- \leqslant \left\|\mu_{k+1}^{\text{IQ}}\right\|_\infty \left\|g_k^-\right\|_1,$$

and we also have

$$\left(\lambda_{k+1}^{\text{IQ}}\right)^{\text{T}} h_k \leqslant \left\|\lambda_{k+1}^{\text{IQ}}\right\|_\infty \|h_k\|_1.$$

By combining these two relations with (3.2a) and (3.4), and by Assumption 3.1(c), we obtain

$$\begin{aligned}\nabla f_k^{\text{T}} d_k^{\text{IQ}} + \frac{1}{2} d_k^{\text{IQ}\text{T}} B_k d_k^{\text{IQ}} &\leqslant \nabla f_k^{\text{T}} d_k^{\text{IQ}} + d_k^{\text{IQ}\text{T}} B_k d_k^{\text{IQ}} \\ &\leqslant \left\|\left(\lambda_{k+1}^{\text{IQ}}, \mu_{k+1}^{\text{IQ}}\right)\right\|_\infty \left(\|h_k\|_1 + \left\|g_k^-\right\|_1\right) \\ &\leqslant \eta(\|h_k\|_1 + \|g_k^-\|_1).\end{aligned}$$

From this relation and (2.10) we have that $\pi_{\text{trial}} \leqslant \eta/(1-\rho)$. Consequently, we have from (2.11) that for all k,

$$\pi_k \leqslant \max\left\{\pi_0, \frac{\eta}{1-\rho} + \pi_b\right\},$$

showing that the sequence of penalty parameters is bounded from above. Finally, note that when the penalty parameter is increased, it is increased by the finite amount π_b. This implies that the sequence of penalty parameters is eventually constant, which completes the proof. □

The next result is crucial.

LEMMA 3.3 Under Assumptions 3.1, Algorithm I yields the limit

$$\lim_{k\to\infty} qred\left(d_k^{\text{IQ}}\right) = 0. \tag{3.5}$$

Proof. By Lemma 3.2, there exists an integer $\bar{k}$ such that for all $k \geqslant \bar{k}$, $\pi_k = \bar{\pi}$. Since for the purpose of proving convergence, the initial finite number of iterations is not relevant, we consider only those iterations with $k \geqslant \bar{k} + 1$.

Let $\mathcal{T}_e$ denote the set of iterates for which the line search along the EQP direction was successful, and hence (2.12) is satisfied. In this case we have

$$\phi_{\bar{\pi}}(x_k) - \phi_{\bar{\pi}}(x_{k+1}) \geqslant \sigma\, qred\left(d_k^{\text{IQ}}\right) \quad \text{for } k \in \mathcal{T}_e. \tag{3.6}$$

Similarly, let $\mathcal{T}_i$ denote the set of iterates for which a backtracking line search along the direction d^{IQ} is performed. In this case we have from (2.14) that

$$\phi_{\bar{\pi}}(x_k) - \phi_{\bar{\pi}}(x_{k+1}) \geqslant \sigma\, \alpha_k qred\left(d_k^{\text{IQ}}\right) \quad \text{for } k \in \mathcal{T}_i. \tag{3.7}$$

By (2.9) we have that $qred(d_k^{\text{IQ}}) \geqslant 0$ for all k and therefore the sequence $\{\phi_{\bar{\pi}}(x_{k+1})\}$ is monotonically decreasing. Since $f(x_k)$ is bounded from below by Assumption 3.1(a) we have that $\{\phi_{\bar{\pi}}(x_k)\}$ is also bounded from below. Thus, $\{\phi_{\bar{\pi}}(x_k)\}$ must converge, i.e.,

$$\lim_{k\to\infty} \left(\phi_{\bar{\pi}}(x_{k+1}) - \phi_{\bar{\pi}}(x_k)\right) = 0,$$

which implies

$$\lim_{k\in\mathcal{T}_e,k\to\infty} qred\left(d_k^{\mathrm{IQ}}\right)=0 \quad \text{and} \quad \lim_{k\in\mathcal{T}_i,k\to\infty} \alpha_k qred\left(d_k^{\mathrm{IQ}}\right)=0. \tag{3.8}$$

If the set $\mathcal{T}_i$ is finite, the limit (3.5) holds since $k\in\mathcal{T}_e$ for all large k. Therefore, we assume that $\mathcal{T}_i$ is infinite and show that Assumptions 3.1, the backtracking line search procedure and the second limit in (3.8) imply that $\lim_{k\in\mathcal{T}_i,k\to\infty} qred(d_k^{\mathrm{IQ}})=0$; this will prove the lemma.

Recalling the definitions (1.4) and (2.6) and Assumption 3.1(a) we have that

$$\begin{aligned}\phi_{\bar\pi}\left(x_k+\alpha_k d_k^{\mathrm{IQ}}\right)-q_{\bar\pi}\left(\alpha_k d_k^{\mathrm{IQ}}\right) &\leqslant f\left(x_k+\alpha_k d_k^{\mathrm{IQ}}\right)-\left(f_k+\alpha_k\nabla f_k^{\mathrm{T}} d_k^{\mathrm{IQ}}\right)\\ &\quad+\bar\pi\left(\left\|h\left(x_k+\alpha_k d_k^{\mathrm{IQ}}\right)\right\|_1-\left\|h_k+\alpha_k H_k d_k^{\mathrm{IQ}}\right\|_1\right)\\ &\quad+\bar\pi\left(\left\|g\left(x_k+\alpha_k d_k^{\mathrm{IQ}}\right)^-\right\|_1-\left\|\left[g_k+\alpha_k G_k d_k^{\mathrm{IQ}}\right]^-\right\|_1\right)\\ &\leqslant L_1\alpha_k^2\left\|d_k^{\mathrm{IQ}}\right\|_2^2,\end{aligned} \tag{3.9}$$

for some constant $L_1>0$ independent of k. From (3.2a) we have

$$d_k^{\mathrm{IQ}}=B_k^{-1}\Big[-\nabla f_k+H_k^{\mathrm{T}}\lambda_{k+1}^{\mathrm{IQ}}+{G_k}^{\mathrm{T}}\mu_{k+1}^{\mathrm{IQ}}\Big].$$

By Assumption 3.1(a) all the terms inside the square brackets are bounded and the matrices B_k are uniformly positive definite. Therefore, for all k, there is a constant L_2 such that

$$\left\|d_k^{\mathrm{IQ}}\right\|_2^2\leqslant L_2,$$

and (3.9) yields

$$\phi_{\bar\pi}\left(x_k+\alpha_k d_k^{\mathrm{IQ}}\right)-q_{\bar\pi}\left(\alpha_k d_k^{\mathrm{IQ}}\right)\leqslant L_1L_2\alpha_k^2. \tag{3.10}$$

The positive definiteness of B_k also implies that $q_{\bar\pi}$ is a convex function, and therefore,

$$q_{\bar\pi}\left(\alpha_k d_k^{\mathrm{IQ}}\right)\leqslant(1-\alpha_k)q_{\bar\pi}(0)+\alpha_k q_{\bar\pi}\left(d_k^{\mathrm{IQ}}\right).$$

Recalling (2.7) this inequality gives

$$q_{\bar\pi}\left(\alpha_k d_k^{\mathrm{IQ}}\right)-q_{\bar\pi}(0)\leqslant-\alpha_k qred\left(d_k^{\mathrm{IQ}}\right).$$

Combining this expression with (3.10), and noting that $\phi_{\bar\pi}(x_k)=q_{\bar\pi}(0)$, we obtain

$$\begin{aligned}\phi_{\bar\pi}\left(x_k+\alpha_k d_k^{\mathrm{IQ}}\right)-\phi_{\bar\pi}(x_k) &=\left[\phi_{\bar\pi}\left(x_k+\alpha_k d_k^{\mathrm{IQ}}\right)-q_{\bar\pi}\left(\alpha_k d_k^{\mathrm{IQ}}\right)\right]+\left[q_{\bar\pi}\left(\alpha_k d_k^{\mathrm{IQ}}\right)-\phi_{\bar\pi}(x_k)\right]\\ &\leqslant-\alpha_k qred\left(d_k^{\mathrm{IQ}}\right)+L_1L_2\alpha_k^2.\end{aligned} \tag{3.11}$$

Now, if

$$\alpha_k\leqslant\frac{(1-\sigma)}{L_1L_2}qred\left(d_k^{\mathrm{IQ}}\right), \tag{3.12}$$

we have

$$\phi_{\bar\pi}\left(x_k+\alpha_k d_k^{\mathrm{IQ}}\right)-\phi_{\bar\pi}(x_k)\leqslant-\alpha_k\sigma qred\left(d_k^{\mathrm{IQ}}\right),$$

and therefore, the sufficient decrease condition (2.14) is satisfied. Since $k \in \mathcal{T}_i$, α_k is chosen by a backtracking line search with a contraction factor τ we conclude that

$$\alpha_k \geqslant \tau \frac{(1-\sigma)}{L_1 L_2} qred\left(d_k^{\text{IQ}}\right).$$

Substituting this inequality in the second limit in (3.8) gives

$$0 = \lim_{k \in \mathcal{T}_i, k \to \infty} \alpha_k qred\left(d_k^{\text{IQ}}\right) \geqslant \lim_{k \in \mathcal{T}_i, k \to \infty} \tau \frac{(1-\sigma)}{L_1 L_2} qred^2\left(d_k^{\text{IQ}}\right) \geqslant 0,$$

which implies that $\lim_{k \in \mathcal{T}_i, k \to \infty} qred(d_k^{\text{IQ}}) = 0$. □

We can now establish some limiting properties of the iterates.

LEMMA 3.4 The sequence $\{x_k\}$ generated by Algorithm I is asymptotically feasible, i.e.,

$$\lim_{k \to \infty} \left(\|h_k\|_1 + \left\|g_k^-\right\|_1\right) = 0. \tag{3.13}$$

Furthermore,

$$\lim_{k \to \infty} g_k^{\text{T}} \mu_{k+1}^{\text{IQ}} = 0 \tag{3.14}$$

and

$$\lim_{k \to \infty} d_k^{\text{IQ}} = 0. \tag{3.15}$$

Proof. From Lemma 3.3 and condition (2.9) we get

$$0 = \lim_{k \to \infty} qred\left(d_k^{\text{IQ}}\right) \geqslant \lim_{k \to \infty} \rho \pi_k \left(\|h_k\|_1 + \left\|g_k^-\right\|_1\right) \geqslant 0.$$

Hence, we have

$$\lim_{k \to \infty} \pi_k \left(\|h_k\|_1 + \left\|g_k^-\right\|_1\right) = 0. \tag{3.16}$$

Because $\{\pi_k\}$ is bounded away from zero, this implies the limit (3.13).

As to (3.14) and (3.15), we first observe from (3.2a) and (3.4) that

$$\begin{aligned} 0 &= \nabla f_k^{\text{T}} d_k^{\text{IQ}} + d_k^{\text{IQT}} B_k d_k^{\text{IQ}} - \left(H_k d_k^{\text{IQ}}\right)^{\text{T}} \lambda_{k+1}^{\text{IQ}} - \left(G_k d_k^{\text{IQ}}\right)^{\text{T}} \mu_{k+1}^{\text{IQ}} \\ &= \nabla f_k^{\text{T}} d_k^{\text{IQ}} + d_k^{\text{IQT}} B_k d_k^{\text{IQ}} + h_k^{\text{T}} \lambda_{k+1}^{\text{IQ}} + g_k^{\text{T}} \mu_{k+1}^{\text{IQ}} \\ &= \left\{ \nabla f_k^{\text{T}} d_k^{\text{IQ}} + \frac{1}{2} d_k^{\text{IQT}} B_k d_k^{\text{IQ}} \right\} + \frac{1}{2} d_k^{\text{IQT}} B_k d_k^{\text{IQ}} + h_k^{\text{T}} \lambda_{k+1}^{\text{IQ}} + g_k^{\text{T}} \mu_{k+1}^{\text{IQ}}. \end{aligned} \tag{3.17}$$

Using the definition (2.8) of $qred$, the limits (3.5) and (3.16), we obtain

$$\lim_{k \to \infty} \nabla f_k^{\text{T}} d_k^{\text{IQ}} + \frac{1}{2} d_k^{\text{IQT}} B_k d_k^{\text{IQ}} = 0, \tag{3.18}$$

and consequently by (3.17)

$$\lim_{k \to \infty} \frac{1}{2} d_k^{\text{IQT}} B_k d_k^{\text{IQ}} + h_k^{\text{T}} \lambda_{k+1}^{\text{IQ}} + g_k^{\text{T}} \mu_{k+1}^{\text{IQ}} = 0. \tag{3.19}$$

By Assumption 3.1(c) the IQP multipliers are bounded and thus from (3.13) we have

$$\lim_{k\to\infty} h_k^{\mathrm{T}} \lambda_{k+1}^{\mathrm{IQ}} = 0. \tag{3.20}$$

Therefore, (3.19) simplifies to

$$\lim_{k\to\infty} \frac{1}{2} d_k^{\mathrm{IQ}\mathrm{T}} B_k d_k^{\mathrm{IQ}} + g_k^{\mathrm{T}} \mu_{k+1}^{\mathrm{IQ}} = 0. \tag{3.21}$$

We also have from (3.13) that $\|g_k^-\|_1 \to 0$, and thus by the boundedness of the IQP multipliers we obtain

$$\lim_{k\to\infty} \sum_{i\in J_k^-} g_i(x_k)\left[\mu_{k+1}^{\mathrm{IQ}}\right]_i = 0, \tag{3.22}$$

where $J_k^- = \{i\colon g_i(x_k) < 0\}$. Therefore,

$$\begin{aligned}\lim_{k\to\infty} g_k^{\mathrm{T}} \mu_{k+1}^{\mathrm{IQ}} &= \lim_{k\to\infty} \left\{ \sum_{i\in J_k^-} g_i(x_k)\left[\mu_{k+1}^{\mathrm{IQ}}\right]_i + \sum_{i\in J_k^+} g_i(x_k)\left[\mu_{k+1}^{\mathrm{IQ}}\right]_i \right\} \\ &= \lim_{k\to\infty} \sum_{i\in J_k^+} g_i(x_k)\left[\mu_{k+1}^{\mathrm{IQ}}\right]_i \\ &\geqslant 0,\end{aligned} \tag{3.23}$$

where $J_k^+ = \{i\colon g_i(x_k) > 0\}$. By (3.1) and (3.23) the limits of both sequences in (3.21) are non-negative, so we conclude that

$$\lim_{k\to\infty} g_k^{\mathrm{T}} \mu_{k+1}^{\mathrm{IQ}} = 0 \quad \text{and} \quad \lim_{k\to\infty} \frac{1}{2} d_k^{\mathrm{IQ}\mathrm{T}} B_k d_k^{\mathrm{IQ}} = 0.$$

Hence, by recalling (3.1) we have

$$0 = \lim_{k\to\infty} \frac{1}{2} d_k^{\mathrm{IQ}\mathrm{T}} B_k d_k^{\mathrm{IQ}} \geqslant \lim_{k\to\infty} M \left\| d_k^{\mathrm{IQ}} \right\|_2^2 \geqslant 0,$$

which implies that $d_k^{\mathrm{IQ}} \to 0$. □

We can now prove the main result of this section.

THEOREM 3.5 Any limit point of the sequence $\{x_k, \lambda_{k+1}^{\mathrm{IQ}}, \mu_{k+1}^{\mathrm{IQ}}\}$ generated by Algorithm I is a KKT point of the nonlinear program (1.1).

Proof. Consider a limit point $\{x_*, \lambda_*, \mu_*\}$ of the sequence $\{x_k, \lambda_{k+1}^{\mathrm{IQ}}, \mu_{k+1}^{\mathrm{IQ}}\}$. Then there is an index set $\mathcal{K}$ such that $(x_k, \lambda_{k+1}^{\mathrm{IQ}}, \mu_{k+1}^{\mathrm{IQ}})_{k\in\mathcal{K}} \to (x_*, \lambda_*, \mu_*)$. Taking limits in (3.2a) we obtain

$$\lim_{k\to\infty, k\in\mathcal{K}} \nabla f_k + B_k d_k^{\mathrm{IQ}} - H_k^{\mathrm{T}} \lambda_{k+1}^{\mathrm{IQ}} - G_k^{\mathrm{T}} \mu_{k+1}^{\mathrm{IQ}} = 0. \tag{3.24}$$

We have shown in Lemma 3.4 that the sequence $\{d_k^{\mathrm{IQ}}\}$ converges to zero, and Assumption 3.1 (d) states that the matrices B_k are bounded, thus

$$\lim_{k\to\infty, k\in\mathcal{K}} B_k d_k^{\mathrm{IQ}} = 0.$$

By Assumption 3.1(a), the functions ∇f, H, G are continuous. Hence, from (3.24),

$$\nabla f(x_*) - H(x_*)^{\mathrm{T}}\lambda_* - G(x_*)^{\mathrm{T}}\mu_* = 0,$$

so that (1.3a) is satisfied. Lemma 3.4 guarantees that x_* is feasible and hence (1.3b) and (1.3c) hold. Condition (1.3e) follows from (3.14), and the non-negativity condition (1.3d) holds by (3.2d). Consequently, the limit point $\{x_*, \lambda_*, \mu_*\}$ satisfies the KKT conditions (1.3). □

This theorem shows that the primal–dual variables $(x_k, \lambda^{\mathrm{IQ}}_{k+1}, \mu^{\mathrm{IQ}}_{k+1})$ can be used to terminate the algorithm. An alternative (primal) stop test would be to terminate the iteration when $\|d^{\mathrm{IQ}}_k\|$ is smaller than a given tolerance. One can also base the convergence test on the new iterate $(x_{k+1}, \lambda_{k+1}, \mu_{k+1})$.

4. Local convergence properties

We now show that Algorithm I identifies the optimal active set once an iterate x_k is close enough to a solution satisfying standard conditions. Furthermore, since for large k the EQP phase computes Newton steps along the optimal active set we show that the rate of convergence of the iteration is quadratic.

We begin by introducing some notation. The working set $\mathcal{W}_k$ is given by (2.2) and the (optimal) active set corresponding to a solution x_* of the nonlinear program (1.1) is defined as

$$\mathcal{W}_* = \{i \in \mathcal{E}\} \cup \{i \in \mathcal{I} | g_i(x_*) = 0\}. \tag{4.1}$$

Recall that $g_{\mathcal{W}} \equiv [g]_{\mathcal{W}\cap\mathcal{I}}$ is the subvector of inequality constraints in the working set $\mathcal{W}$. We denote by $c_{\mathcal{W}}$ the vector of constraints in the working set, i.e.,

$$c_{\mathcal{W}}(x_k) = \begin{bmatrix} h(x_k) \\ g_{\mathcal{W}}(x_k) \end{bmatrix}, \tag{4.2}$$

and denote its Jacobian by $A_{\mathcal{W}}$, specifically

$$A_{\mathcal{W}}(x_k) = \begin{bmatrix} H(x_k) \\ G_{\mathcal{W}}(x_k) \end{bmatrix}, \quad \text{with } G_{\mathcal{W}}(x_k) = \left[\nabla g_i(x_k)^{\mathrm{T}}\right]_{i\in\mathcal{W}\cap\mathcal{I}}. \tag{4.3}$$

Our local convergence results are established under the following assumptions.

ASSUMPTION 4.1 Let (x_*, λ_*, μ_*) be a primal–dual solution to (1.1) and let $\mathcal{W}_*$ be the corresponding optimal active set.

(a) Smoothness. In a neighbourhood of x_*, the functions f, g and h are twice continuously differentiable with Lipschitz continuous second derivatives.
(b) Regularity. The Jacobian of the active constraints at x_*, denoted by $A_{\mathcal{W}_*}(x_*)$, has full row rank.
(c) Strict complementarity. $\mu_* + g(x_*) > 0$.
(d) Regularity of IQP model. For all k the subproblem (2.1) is feasible, and the matrices B_k are positive definite and their smallest eigenvalue is bounded away from zero.
(e) Second-order sufficiency. The Hessian W defined in (2.4) satisfies $y^{\mathrm{T}}W(x_*, \lambda_*, \mu_*)y > 0$ for all $y \neq 0$ such that $A_{\mathcal{W}_*}(x_*)y = 0$.

Throughout the analysis we assume that $\|\cdot\|$ denotes the 2-norm, unless indicated otherwise. The following lemma, which was first proved by Robinson (1974), states that near a solution x_*, the IQP step identifies the optimal active set $\mathcal{W}_*$. We give a proof of this result because some of the arguments are used again in the proof of Theorem 4.3.

LEMMA 4.2 Suppose that Assumptions 4.1 hold. If x_k is sufficiently close to x_*, the working set (2.2) identified by the solution of the IQP subproblem (2.1) is the optimal active set, i.e., $\mathcal{W}_k = \mathcal{W}_*$.

Proof. By Assumption 4.1(d), the IQP subproblem (2.1) has a unique primal optimal solution d_k^{IQ}, and hence the working set $\mathcal{W}_k$ is uniquely determined by (2.2). Let us recall definitions (4.2) and (4.3) and consider the system

$$\begin{bmatrix} B_k & -A_{\mathcal{W}_*}^{\mathrm{T}}(x_k) \\ A_{\mathcal{W}_*}(x_k) & 0 \end{bmatrix} \begin{bmatrix} d \\ \gamma \end{bmatrix} = - \begin{bmatrix} \nabla f_k \\ c_{\mathcal{W}_*}(x_k) \end{bmatrix}, \tag{4.4}$$

which is defined in terms of the optimal active set $\mathcal{W}_*$. We now show that when x_k is close to x_*, the IQP step can be computed via solution of the system (4.4).

Let us define the neighbourhood

$$\mathcal{N}_*(\epsilon) \triangleq \{x \,|\, \|x - x_*\| \leqslant \epsilon\} \quad \text{with } \epsilon > 0. \tag{4.5}$$

By Assumptions 4.1(a), (b), (d), for ϵ sufficiently small and $x_k \in \mathcal{N}_*(\epsilon)$, the linear system (4.4) is nonsingular and the inverse of its coefficient matrix is bounded above in norm by some constant $\delta_1 > 0$. Let us define

$$\gamma_* = \begin{bmatrix} \lambda_* \\ [\mu_*]_{\mathcal{W}_*} \end{bmatrix}. \tag{4.6}$$

From (4.4) we have that

$$\begin{bmatrix} d \\ \gamma - \gamma_* \end{bmatrix} = - \begin{bmatrix} B_k & -A_{\mathcal{W}_*}^{\mathrm{T}}(x_k) \\ A_{\mathcal{W}_*}(x_k) & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nabla f_k - A_{\mathcal{W}_*}^{\mathrm{T}}(x_k)\gamma_* \\ c_{\mathcal{W}_*}(x_k) \end{bmatrix}, \tag{4.7}$$

and hence

$$\left\| \begin{bmatrix} d \\ \gamma - \gamma_* \end{bmatrix} \right\| \leqslant \delta_1 \left\| \begin{bmatrix} \nabla f_k - A_{\mathcal{W}_*}^{\mathrm{T}}(x_k)\gamma_* \\ c_{\mathcal{W}_*}(x_k) \end{bmatrix} \right\|. \tag{4.8}$$

Furthermore, Assumption 4.1(a) imply that ∇f, $c_{\mathcal{W}_*}$, $A_{\mathcal{W}_*}$ are Lipschitz continuous and therefore for all $x_k \in \mathcal{N}_*(\epsilon)$ there exist constants κ_f, κ_c and κ_a such that

$$\begin{aligned} \|\nabla f_k - \nabla f(x_*)\| &\leqslant \kappa_f \|x_k - x_*\| \leqslant \kappa_f \epsilon, \\ \|c_{\mathcal{W}_*}(x_k) - c_{\mathcal{W}_*}(x_*)\| &\leqslant \kappa_c \|x_k - x_*\| \leqslant \kappa_c \epsilon, \\ \|A_{\mathcal{W}_*}(x_k) - A_{\mathcal{W}_*}(x_*)\| &\leqslant \kappa_a \|x_k - x_*\| \leqslant \kappa_a \epsilon. \end{aligned} \tag{4.9}$$

By (4.6) we can express the KKT conditions (1.3) of the nonlinear program (1.1) as

$$\begin{bmatrix} \nabla f(x_*) - A_{\mathcal{W}_*}^{\mathrm{T}}(x_*)\gamma_* \\ c_{\mathcal{W}_*}(x_*) \end{bmatrix} = 0, \tag{4.10}$$

together with

$$[\mu_*]_{\mathcal{W}_*} > 0, \quad [\mu_*]_{\mathcal{W}_*^{\mathrm{c}}} = 0, \quad g_{\mathcal{W}_*^{\mathrm{c}}}(x_*) > 0, \tag{4.11}$$

where condition (4.11) follows from Assumption 4.1(c). Therefore, we have from (4.10) and (4.9) that

$$\begin{aligned}\left\|\begin{bmatrix}\nabla f_k - A_{\mathcal{W}_*}^{\mathrm{T}}(x_k)\gamma_* \\ c_{\mathcal{W}_*}(x_k)\end{bmatrix}\right\| &= \left\|\begin{bmatrix}(\nabla f_k - \nabla f(x_*)) - (A_{\mathcal{W}_*}(x_k) - A_{\mathcal{W}_*}(x_*))^{\mathrm{T}}\gamma_* \\ c_{\mathcal{W}_*}(x_k) - c_{\mathcal{W}_*}(x_*)\end{bmatrix}\right\| \\ &\leqslant \|\nabla f_k - \nabla f(x_*)\| + \|c_{\mathcal{W}_*}(x_k) - c_{\mathcal{W}_*}(x_*)\| + \|A_{\mathcal{W}_*}(x_k) - A_{\mathcal{W}_*}(x_*)\|\|\gamma_*\| \\ &\leqslant (\kappa_f + \kappa_c + \kappa_a\|\gamma_*\|)\|x_k - x_*\|.\end{aligned} \tag{4.12}$$

If we define $\kappa_o = \kappa_f + \kappa_c + \kappa_a\|\gamma_*\|$, we obtain from (4.8) that

$$\left\|\begin{bmatrix} d \\ \gamma - \gamma_*\end{bmatrix}\right\| \leqslant \kappa_o\delta_1\|x_k - x_*\| \leqslant \kappa_o\delta_1\epsilon. \tag{4.13}$$

Let us now write γ in terms of components whose dimensions are compatible with those of γ_* in (4.6), i.e.,

$$\gamma = \begin{bmatrix}\lambda \\ \nu\end{bmatrix}. \tag{4.14}$$

Then, from (4.13),

$$\|d\| \leqslant \kappa_o\delta_1\epsilon \quad \text{and} \quad \left\|\nu - [\mu_*]_{\mathcal{W}_*}\right\| \leqslant \kappa_o\delta_1\epsilon.$$

Hence,

$$\begin{aligned}\nu &= [\mu_*]_{\mathcal{W}_*} - \left([\mu_*]_{\mathcal{W}_*} - \nu\right) \\ &\geqslant [\mu_*]_{\mathcal{W}_*} - \kappa_o\delta_1\epsilon\,\mathbf{1},\end{aligned} \tag{4.15}$$

and

$$\begin{aligned}g_{\mathcal{W}_*^{\mathrm{c}}}(x_k) + G_{\mathcal{W}_*^{\mathrm{c}}}(x_k)d &= g_{\mathcal{W}_*^{\mathrm{c}}}(x_*) - \left(g_{\mathcal{W}_*^{\mathrm{c}}}(x_*) - g_{\mathcal{W}_*^{\mathrm{c}}}(x_k)\right) \\ &\quad - \left(G_{\mathcal{W}_*^{\mathrm{c}}}(x_*) - G_{\mathcal{W}_*^{\mathrm{c}}}(x_k)\right)d + G_{\mathcal{W}_*^{\mathrm{c}}}(x_*)d \\ &\geqslant g_{\mathcal{W}_*^{\mathrm{c}}}(x_*) - \left(\kappa_g + \kappa_G\kappa_o\delta_1\epsilon + \|G_{\mathcal{W}_*^{\mathrm{c}}}(x_*)\|\kappa_o\delta_1\right)\epsilon\,\mathbf{1},\end{aligned} \tag{4.16}$$

where κ_g and κ_G are, respectively, Lipschitz constants for $g_{\mathcal{W}_*^{\mathrm{c}}}(x_*)$ and $G_{\mathcal{W}_*^{\mathrm{c}}}(x)$ over $\mathcal{N}_*(\epsilon)$ and $\mathbf{1}$ is a vector of all ones of appropriate dimension. Therefore, if ϵ is small enough, we have from (4.11), (4.15), (4.16) that

$$\nu > 0, \quad g_{\mathcal{W}_*^{\mathrm{c}}}(x_k) + G_{\mathcal{W}_*^{\mathrm{c}}}(x_k)d > 0. \tag{4.17}$$

We can now construct the solution of the IQP subproblem as follows:

$$d_k^{\mathrm{IQ}} = d, \quad \begin{bmatrix}\lambda_{k+1}^{\mathrm{IQ}} \\ \left[\mu_{k+1}^{\mathrm{IQ}}\right]_{\mathcal{W}_*}\end{bmatrix} = \gamma = \begin{bmatrix}\lambda \\ \nu\end{bmatrix}, \quad \left[\mu_{k+1}^{\mathrm{IQ}}\right]_{\mathcal{W}_*^{\mathrm{c}}} = 0. \tag{4.18}$$

By (4.4) and (4.17) it is easy to verify that this primal–dual solution satisfies the KKT conditions (3.2) of the IQP subproblem. Therefore, the vector d_k^{IQ} constructed in this manner is indeed the unique primal optimal solution of the IQP subproblem and its working set satisfies $\mathcal{W}_k = \mathcal{W}_*$. □

We now present the main result of this section. It shows that, if the algorithm takes the full step for all x_k sufficiently close to x_*, the iteration is quadratically convergent.

THEOREM 4.3 Suppose that: (i) Assumptions 4.1 hold; (ii) For x_k sufficiently close to x_*, Algorithm I computes the step d_k by (2.13) with $\alpha_k = 1$. Then, there exists a neighbourhood $\mathcal{N}_{**}(\epsilon)$ such that if $x_k \in \mathcal{N}_{**}(\epsilon)$, the sequence $\{(x_l, \lambda_l, \mu_l)\}_{l=k}^{\infty}$ converges quadratically to (x_*, λ_*, μ_*).

Proof. By Lemma 4.2, if x_k is sufficiently close to x_*, we have $\mathcal{W}_k = \mathcal{W}_*$. The KKT conditions of the EQP subproblem (2.3) are therefore given by

$$\begin{bmatrix} W_k & -A_{\mathcal{W}_*}^{\mathrm{T}}(x_k) \\ A_{\mathcal{W}_*}(x_k) & 0 \end{bmatrix} \begin{bmatrix} d + d_k^{\mathrm{IQ}} \\ \theta \end{bmatrix} = - \begin{bmatrix} \nabla f_k \\ c_{\mathcal{W}_*}(x_k) \end{bmatrix}. \tag{4.19}$$

Since, by Assumptions 4.1(b), (e), the matrix

$$\begin{bmatrix} W(x, \lambda, \mu) & -A_{\mathcal{W}_*}^{\mathrm{T}}(x) \\ A_{\mathcal{W}_*}(x) & 0 \end{bmatrix}$$

is nonsingular at (x_*, λ_*, μ_*), we have by (2.4), (4.13), (4.18) and Assumption 4.1(a) that, if x_k is sufficiently close to x_*, the coefficient matrix of (4.19) is also nonsingular, whose inverse is bounded above in norm. Hence, the solution to (4.19) is given by

$$(d, \theta) = \left(d_k^{\mathrm{EQ}}, \theta_{k+1}^{\mathrm{EQ}}\right) \quad \text{with} \quad \theta_{k+1}^{\mathrm{EQ}} \triangleq \begin{bmatrix} \lambda_{k+1}^{\mathrm{EQ}} \\ \mu_{k+1}^{\mathrm{EQ}} \end{bmatrix}. \tag{4.20}$$

Note that (4.19) has the same form as (4.4), except that B_k is replaced by W_k. Thus, we can follow the same reasoning as in the proof of Lemma 4.2 and deduce that $(d_k^{\mathrm{EQ}} + d_k^{\mathrm{IQ}}, \theta_{k+1}^{\mathrm{EQ}})$ also satisfies (4.17), i.e.,

$$\mu_{k+1}^{\mathrm{EQ}} > 0 \quad \text{and} \quad g_{\mathcal{W}_*^{\mathrm{c}}}(x_k) + G_{\mathcal{W}_*^{\mathrm{c}}}(x_k)\left(d_k^{\mathrm{IQ}} + d_k^{\mathrm{EQ}}\right) > 0. \tag{4.21}$$

As a result, for x_k sufficiently close to x_*, Step 4 of Algorithm I will choose $\beta = 1$, Step 6 will set

$$d_k = d_k^{\mathrm{IQ}} + d_k^{\mathrm{EQ}} \quad \text{and} \quad x_{k+1} - x_k = d_k^{\mathrm{IQ}} + d_k^{\mathrm{EQ}}, \tag{4.22}$$

and Step 7 will set

$$\lambda_{k+1} = \lambda_{k+1}^{\mathrm{EQ}}, \quad [\mu_{k+1}]_{\mathcal{W}_*} = \mu_{k+1}^{\mathrm{EQ}}, \quad [\mu_{k+1}]_{\mathcal{W}_*^{\mathrm{c}}} = 0. \tag{4.23}$$

Let us define a matrix function as

$$\hat{W}(x, \theta) \triangleq \nabla^2 f(x) - \sum_{i \in \mathcal{E}} \theta_i \nabla^2 h_i(x) - \sum_{i \in \mathcal{I} \cap \mathcal{W}_*} \theta_i \nabla^2 g_i(x). \tag{4.24}$$

Then, by recalling (2.4) and plugging (4.20), (4.22) into (4.19), we further deduce that the EQP iterate $(x_{k+1}, \theta_{k+1}^{\mathrm{EQP}})$ satisfies

$$\begin{bmatrix} \hat{W}(x_k, \theta_{k+1}^{\mathrm{IQ}}) & -A_{\mathcal{W}_*}^{\mathrm{T}}(x_k) \\ A_{\mathcal{W}_*}(x_k) & 0 \end{bmatrix} \begin{bmatrix} x_{k+1} - x_k \\ \theta_{k+1}^{\mathrm{EQ}} - \theta_{k+1}^{\mathrm{IQ}} \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) - A_{\mathcal{W}_*}^{\mathrm{T}}(x_k)\theta_{k+1}^{\mathrm{IQ}} \\ c_{\mathcal{W}_*}(x_k) \end{bmatrix}, \tag{4.25}$$

where

$$\theta_{k+1}^{\mathrm{IQ}} \triangleq \begin{bmatrix} \lambda_{k+1}^{\mathrm{IQ}} \\ \left[\mu_{k+1}^{\mathrm{IQ}}\right]_{\mathcal{W}_*} \end{bmatrix} \tag{4.26}$$

was the vector of IQP multipliers that has already been computed. Note that the system (4.25) is the Newton iteration applied to the nonlinear system

$$\nabla f(x) - A^{\mathrm{T}}_{\mathcal{W}_*}(x)\theta = 0, \tag{4.27a}$$

$$c_{\mathcal{W}_*}(x) = 0 \tag{4.27b}$$

at the point $(x_k, \theta^{\mathrm{IQ}}_{k+1})$ for which the solution is $(x_{k+1}, \theta^{\mathrm{EQ}}_{k+1})$. Thus, we can perform standard Newton analysis to establish quadratic convergence. We first observe that, by Assumption 4.1(a), the matrix

$$F(x,\theta) \triangleq \begin{bmatrix} \hat{W}(x,\theta) & -A^{\mathrm{T}}_{\mathcal{W}_*}(x) \\ A_{\mathcal{W}_*}(x) & 0 \end{bmatrix}$$

is Lipschitz continuous with Lipschitz constant κ_1. If we define

$$\theta_* = \begin{bmatrix} \lambda_* \\ [\mu_*]_{\mathcal{W}_*} \end{bmatrix}, \tag{4.28}$$

then, as we have noted for the linear system (4.19), in a neighbourhood of (x_*, θ_*), F is invertible and F^{-1} is bounded above in norm by a constant κ_2. Thus, by Theorem 5.2.1 of Dennis & Schnabel (1983), if $(x_k, \theta^{\mathrm{IQ}}_{k+1})$ satisfies

$$\left\| \begin{bmatrix} x_k - x_* \\ \theta^{\mathrm{IQ}}_{k+1} - \theta_* \end{bmatrix} \right\| \leqslant \frac{1}{2\kappa_1\kappa_2}, \tag{4.29}$$

then

$$\left\| \begin{bmatrix} x_{k+1} - x_* \\ \theta^{\mathrm{EQ}}_{k+1} - \theta_* \end{bmatrix} \right\| \leqslant \kappa_1\kappa_2 \left\| \begin{bmatrix} x_k - x_* \\ \theta^{\mathrm{IQ}}_{k+1} - \theta_* \end{bmatrix} \right\|^2. \tag{4.30}$$

Let us define the neighbourhood

$$\mathcal{N}_{**}(\epsilon) = \{x \,|\, \|x - x_*\| \leqslant \epsilon\},$$

where $\epsilon > 0$ is small enough that all the properties derived so far in this proof and the proof of Lemma 4.2 hold for $x_k \in \mathcal{N}_{**}(\epsilon)$. Then, from (4.13) and (4.18), there exists a constant $\kappa_3 > 1$ such that $x_k \in \mathcal{N}_{**}(\epsilon)$ implies $\|\theta^{\mathrm{IQ}}_{k+1} - \theta_*\| < \kappa_3 \|x_k - x_*\|$, which further implies that

$$\left\| \begin{bmatrix} x_k - x_* \\ \theta^{\mathrm{IQ}}_{k+1} - \theta_* \end{bmatrix} \right\| \leqslant \sqrt{1 + \kappa_3^2}\ \|x_k - x_*\|, \tag{4.31}$$

and that (4.29) and (4.30) are satisfied with sufficiently small ϵ, as desired. Hence, it follows from (4.23), (4.30), (4.31) and the definitions in (4.20), (4.28) that if $x_k \in \mathcal{N}_{**}(\epsilon)$,

$$\left\| \begin{bmatrix} x_{k+1} - x_* \\ \lambda_{k+1} - \lambda_* \\ \mu_{k+1} - \mu_* \end{bmatrix} \right\| \leqslant \kappa_1\kappa_2 \left(1 + \kappa_3^2\right) \|x_k - x_*\|^2 \leqslant \kappa_1\kappa_2 \left(1 + \kappa_3^2\right) \left\| \begin{bmatrix} x_k - x_* \\ \lambda_k - \lambda_* \\ \mu_k - \mu_* \end{bmatrix} \right\|^2. \tag{4.32}$$

Let ϵ be sufficiently small such that $\epsilon \leqslant \min\{1,\ 1/\kappa_1\kappa_2(1 + \kappa_3^2)\}$, then $x_{k+1} \in \mathcal{N}_{**}(\epsilon)$. Therefore, we have shown that if $x_k \in \mathcal{N}_{**}(\epsilon)$, all subsequent iterates remain in $\mathcal{N}_{**}(\epsilon)$ and converge quadratically to (x_*, λ_*, μ_*). □

This quadratic convergence result is more satisfying than those established in the literature of SQP methods that use exact Hessians. This is because, even in a neighbourhood of a solution satisfying Assumptions 4.1, the quadratic program (2.1) may have several local minimizers if B_k is replaced by the Hessian W_k. Thus, for classical SQP methods it is necessary to assume that the quadratic programming solver selects a local minimizer with certain properties; for example, the one closest to the current iterate. Our quadratic convergence result relies only on the second-order sufficiency conditions of the problem.

To establish Theorem 4.3 we have assumed that near the solution the line search condition (2.12) is satisfied for $\alpha_k = 1$. As is well known, however, the nonsmooth penalty function ϕ_π may reject unit step lengths (the Maratos effect) and some additional features must be incorporated into the algorithm to prevent this from happening. They include second-order corrections or watchdog steps; see, e.g., Nocedal & Wright (2006).

5. Implementation and numerical results

The SQP+ algorithm can be implemented as an extension of any SQP method that solves convex quadratic programs for the step computation. Our implementation is based on SQPlab, a MATLAB package developed by Gilbert (2007) that offers a classical line search SQP method using an ℓ_1 merit function and Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton updating. We chose SQPlab as a platform for our prototype implementation of the SQP+ method because it allows us to easily evaluate various components of the algorithm in a controlled setting.

We made two modifications to SQPlab in order to improve its performance. Instead of using MathWorks' routine quadprog to solve the quadratic program (2.1), we employed KNITRO/ACTIVE (Byrd *et al.*, 2006) for this task. Unlike quadprog, the KNITRO/ACTIVE code had no difficulties solving the quadratic subproblems generated during the IQP phase and provided reliable multiplier estimates. The second modification concerns the choice of the penalty parameter π, which in SQPlab is based on the size of Lagrange multiplier estimates. We replaced this technique by the rule (2.10)–(2.11) that is known to perform better in practice.

The BFGS quasi-Newton approximation B_k used in the IQP phase of the SQP+ method is updated using information from the full step. We define the correction pairs as

$$s_k = x_{k+1} - x_k \quad \text{and} \quad y_k = \nabla_x L(x_{k+1}, \lambda_{k+1}, \mu_{k+1}) - \nabla_x L(x_k, \lambda_{k+1}, \mu_{k+1}),$$

where λ_{k+1} is given in Step 7 of Algorithm I, and modify y_k, if necessary, by means of Powell's damping procedure (Powell, 1978) to ensure that all Hessian approximations B_k are positive definite.

As mentioned in Section 2 we can compute an approximate solution of the EQP problem (2.3) by either applying the projected conjugate gradient method (Nocedal & Wright, 2006) (and adding a trust region constraint to (2.3)) or using a direct method. Here we follow the latter approach. When the EQP problem (2.3) is convex, its solution solves the KKT system

$$\begin{bmatrix} W_k & H^{\mathrm{T}}(x_k) & G^{\mathrm{T}}_{\mathcal{W}_k}(x_k) \\ H(x_k) & 0 & 0 \\ G_{\mathcal{W}_k}(x_k) & 0 & 0 \end{bmatrix} \begin{bmatrix} d^{\mathrm{EQ}} \\ \lambda^{\mathrm{EQ}} \\ \mu^{\mathrm{EQ}} \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) + W_k d^{\mathrm{IQ}} \\ 0 \\ 0 \end{bmatrix}, \tag{5.1}$$

where, as in Section 4, $G_{\mathcal{W}_k}$ is the matrix obtained by selecting the rows of G corresponding to the elements of $\mathcal{W}_k$. If the inertia of the KKT matrix in (5.1) is given by

$$(n, |\mathcal{W}_k|, 0), \tag{5.2}$$

then we know that problem (2.3) has a unique minimizer (Nocedal & Wright, 2006). If this is not the case (which can occur also if the gradients of the constraints in the working set $\mathcal{W}_k$ are not linearly independent), we simply skip the EQP step. (We also tested the option of adding a sufficiently large multiple of the identity matrix to W_k, as discussed in Vanderbei & Shanno, 1999, so that the inertia of the modified system is given by (5.2) but such an EQP step did not yield an overall improvement in performance.)

In the line search procedure we test the unit step length along the full step $d^{\text{IQ}} + \beta d^{\text{EQ}}$. If $\alpha_k = 1$ does not yield the sufficient decrease condition (2.12), we fall back on the IQP step and commence the backtracking line search from there. The algorithm terminates when the following conditions are satisfied

$$\begin{aligned} \left\| \nabla f(x_k) - H(x_k)^{\mathrm{T}} \lambda_k - G(x_k)^{\mathrm{T}} \mu_k \right\|_2 &\leqslant \epsilon_1, \\ \|(h(x_k), \max[0, -g(x_k)])\|_2 &\leqslant \epsilon_2, \\ \left\| \mu_k^{\mathrm{T}} g(x_k) \right\|_2 &\leqslant \epsilon_3 \end{aligned} \tag{5.3}$$

for some constants $\epsilon_1, \epsilon_2, \epsilon_3$. Following is a detailed description of the algorithm used in our tests.

Algorithm SQP+

Initial data: $x_0 \in \mathbb{R}^n$; $\pi_0, \pi_b, \epsilon_1, \epsilon_2, \epsilon_3 > 0$; $\sigma, \rho, \tau \in (0, 1)$; $B_1 = I$.

For $k = 1, 2, \ldots$ until (x_k, λ_k, μ_k) satisfies the termination test (5.3):

1. Compute the step d_k^{IQ} and multipliers $\lambda_{k+1}^{\text{IQ}}$, μ_{k+1}^{IQ} by solving (2.1);
2. Determine the working set $\mathcal{W}_k$. If $|\mathcal{W}_k| \geqslant n$, go to Step 0g?
3. Factor the system (5.1); if its inertia is not given by (5.2), then go to Step 7. Else, compute the EQP step d_k^{EQ} and multipliers $\lambda_{k+1}^{\text{EQ}}$, μ_{k+1}^{EQ} by solving the linear system (5.1);
4. Compute $\beta \geqslant 0$ to be the largest number in $[0, 1]$ such that $d_k = d_k^{\text{IQ}} + \beta d_k^{\text{EQ}}$ satisfies (2.1b) and (2.1c);
5. Update the penalty parameter π_k by the rule (2.10)–(2.11);
6. If the sufficient decrease condition (2.12) is satisfied, for $\alpha_k = 1$, then set

$$x_{k+1} = x_k + d_k^{\text{IQ}} + \beta d_k^{\text{EQ}},$$

$$\lambda_{k+1} = \lambda_{k+1}^{\text{EQ}}, \quad [\mu_{k+1}]_{\mathcal{W}_k} = \max\left(0, \mu_{k+1}^{\text{EQ}}\right), \quad [\mu_{k+1}]_{\mathcal{W}_k^{\text{c}}} = 0,$$

   and go to Step 8;
7. Let $\alpha_k \in (0, 1]$ be the first member of the sequence $\{1, \tau, \tau^2, \ldots\}$ such that

$$\phi_{\pi_k}\left(x_k + \alpha_k d_k^{\text{IQ}}\right) \leqslant \phi_{\pi_k}(x_k) - \sigma \alpha_k qred\left(d_k^{\text{IQ}}\right), \tag{5.4}$$

   and set

$$x_{k+1} = x_k + \alpha_k d_k^{\text{IQ}}; \quad (\lambda_{k+1}, \mu_{k+1}) = \left((1 - \alpha_k)\lambda_k + \alpha_k \lambda^{\text{IQ}}, (1 - \alpha_k)\mu_k + \alpha_k \mu^{\text{IQ}}\right); \tag{5.5}$$

8. Update B_{k+1} from B_k using the BFGS method with Powell damping.

The constants in the algorithm are chosen as follows: $\epsilon_1 = \epsilon_2 = \epsilon_3 = 10^{-5}$, $\sigma = 10^{-4}$, $\tau = 0.5$ and $\rho = 0.1$. Initially, $\pi_b = 0$ and at the end of the first iteration it is reset to $\pi_b = \max(\sqrt{\epsilon_m}, \|(\lambda_1, \mu_1)\|_\infty/100)$.

In Section 5.1 we compare Algorithm SQP+ with a *classical SQP method* that is obtained by omitting Steps 2–4 and 6 in Algorithm SQP+. We will refer to this classical SQP method as C-SQP.

5.1 *Numerical experiments*

Our test set consists of 167 small- and medium-size problems from the CUTEr (Gould *et al.*, 2003) collection. The names of problems and their characteristics are given in Tables A1–A4 of the Appendix. We did not test large problems because our MATLAB code is not suitable for large-scale applications. We believe, however, that the merits of the SQP+ approach can be evaluated in the context of small- and medium-size problems because, as we discuss below, the EQP phase does not add a significant cost to the iteration.

Our implementation does not include a feature to ensure that the constraints (2.1b)–(2.1c) in the SQP subproblem are always feasible. Therefore, those problems for which SQP+ or C-SQP encountered an infeasible subproblem were removed from the test set.

The performance of SQP+ and the classical SQP method (both implemented in our modification of SQPlab) is reported in Fig. 1. This figure plots the logarithmic performance profiles of Dolan & Moré (2002) based on the number of iterations required for convergence. (A comparison based on number of function evaluations would give very similar performance profiles.) Detailed results are given in Tables A1–A4. (In only seven of the test problems was the EQP phase skipped because the inertia was not given by (5.2).)

The benefit of adding the EQP phase can be observed by comparing SQP+ and C-SQP since the other aspects of these two algorithms are identical. The results presented here suggest that this benefit can be substantial.

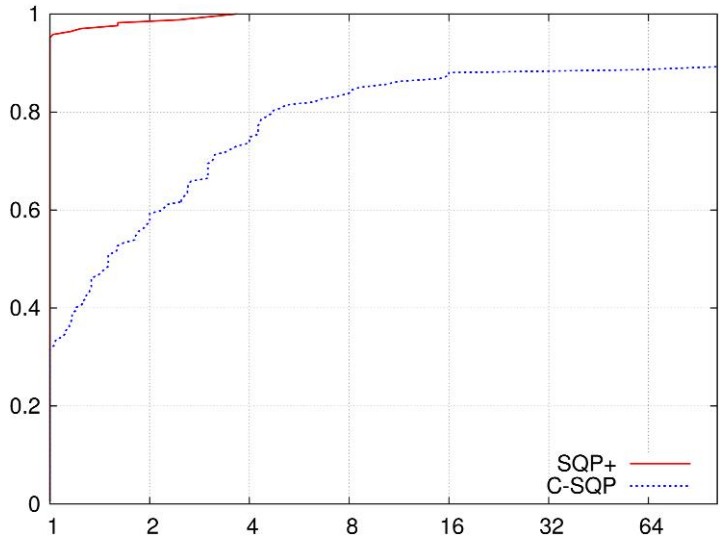

FIG. 1. Comparison of C-SQP and SQP+ in terms of iterations.

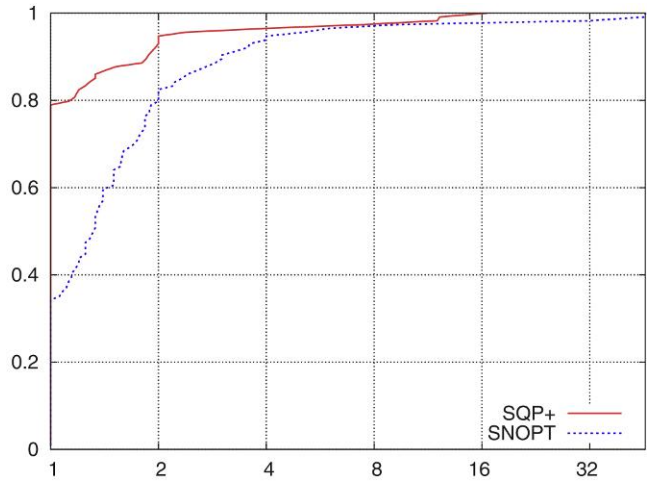

FIG. 2. Comparison of the number of major iterations in SNOPT and SQP+.

To obtain another measure of the performance of the SQP+ algorithm we also compare it with SNOPT (Gill *et al.*, 2002, 2005) version 7.2-1 in terms of major iterations. We used the same test set as in the previous experiment except that we removed all linear and quadratic programs because SNOPT recognizes their structure and solves them in one major iteration (e.g., using the exact Hessian for a quadratic program), whereas the SQP+ method treats linear and quadratic programs as general nonlinear programs. The results are reported in Fig. 2. Although the test set used for this experiment is limited, we believe that the results are encouraging both in terms of robustness and number of iterations.

As already mentioned our MATLAB implementation does not permit timing experiments on large-scale problems, and therefore an evaluation of the speed of the SQP+ approach must await the development of a sparse large-scale implementation. We mention, nonetheless, that the additional computational cost incurred by the EQP phase is not likely to be significant. This view is based on the numerical experiments reported in Byrd *et al.* (2004) using an SLQP method that contains an EQP phase, just like Algorithm SQP+. That paper reports that the cost of the EQP phase is dominated by the cost of the linear programming phase. We can expect the same in the SQP+ algorithm since solving a quadratic program is generally more expensive than solving a linear program.

We conclude this section by describing an enhancement to the EQP phase that we developed and tested. Rather than using the contraction parameter β in Step 4 of Algorithm SQP+ to ensure that all linearized constraints are satisfied, we experimented with the option of computing the minimum norm projection of the step $d^{\mathrm{IQ}} + d^{\mathrm{EQ}}$ onto the feasible region defined by (2.1b) and (2.1c). We found that the improvements in performance obtained with the projection approach were not substantial except on bound constrained problems. Since computing the projection is expensive this option does not seem viable at this point and further investigation is needed to determine the most effective implementation of the EQP phase.

6. Final remarks

We have presented an SQP method that can employ exact second derivative information but never solves indefinite inequality-constrained quadratic programs. It is a two-stage method in which global

convergence and active-set identification are driven by an IQP phase (which solves convex quadratic programs), while fast asymptotic convergence is achieved by an EQP phase (which solves equality constrained subproblems). The numerical results presented in this paper suggest that the addition of the EQP phase leads to an important decrease in the total number of iterations and function evaluations.

Simultaneously with this work, Gould & Robinson (2010b) have developed a trust region method that has many similarities with the algorithm proposed here. The main difference between their approach and ours lies in the way the IQP and EQP steps are combined, in their use of trust regions, in the procedure for computing the EQP step, in the definition of Cauchy decrease and in various details of implementation.

The approach presented here does not overcome one of the main limitations of SQP methods, namely the major computational cost of solving large quadratic programs, particularly when the reduced space is large. Although the IQP matrix B_k can be chosen to be a simple matrix such as a limited memory BFGS matrix, or even a diagonal matrix, the cost of the IQP phase can still be significant. Therefore, we view SQP+ as a method that is applicable to the class of problems that is currently solved by SQP methods.

Acknowledgement

We, the authors, would like to thank Richard Byrd and one of the referees for many useful comments.

Funding

Asociación Mexicana de Cultura AC and CONACyT–National Science Foundation (NSF) (J110.388/2006 to J.L.M.); Department of Energy (DE-FG02-87ER25047 to J.N.) and NSF (DMS-0810213 to Y.W.).

References

BOGGS, P. T. & TOLLE, J. W. (1995) Sequential quadratic programming. *Acta Numerica*, **4**, 1–51.

BONNANS, J. F. & LAUNAY, G. (1995) Sequential quadratic-programming with penalization of the displacement. *SIAM J. Optim.*, **5**, 792–812.

BURKE, J. V. & HAN, S. P. (1989) A robust sequential quadratic-programming method. *Math. Program.*, **43**, 277–303.

BYRD, R. H., GOULD, N. I. M., NOCEDAL, J. & WALTZ, R. A. (2004) An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. *Math. Program. Ser. B*, **100**, 27–48.

BYRD, R. H., GOULD, N. I. M., NOCEDAL, J. & WALTZ, R. A. (2006) On the convergence of successive linear-quadratic programming algorithms. *SIAM J. Optim.*, **16**, 471–489.

BYRD, R. H., NOCEDAL, J. & LÓPEZ-CALVA, G. (2009) A line search exact penalty method using steering rules. *Technical Report*. Evanston: Optimization Center, Northwestern University.

BYRD, R. H., NOCEDAL, J. & WALTZ, R. A. (2006) KNITRO: an integrated package for nonlinear optimization. *Large-Scale Nonlinear Optimization* (G. di Pillo & M. Roma eds). Springer, pp. 35–59.

BYRD, R. H., NOCEDAL, J. & WALTZ, R. A. (2008) Steering exact penalty methods. *Optim. Methods Softw.*, **23**, 197–213.

CHAMBERLAIN, R. M., POWELL, M. J. D., LEMARÉCHAL, C. & PEDERSEN, H. C. (1982) The watchdog technique for forcing convergence in algorithms for constrained optimization. *Math. Program. Stud.*, **16**, 1–17.

CHEN, L. & GOLDFARB, D. (2006) Interior-point ℓ_2 penalty methods for nonlinear programming with strong global convergence properties. *Math. Program.*, **108**, 1–36.

CHIN, C. M. & FLETCHER, R. (2003) On the global convergence of an SLP-filter algorithm that takes EQP steps. *Math. Program. Ser. A*, **96**, 161–177.

COLEMAN, T. F. & VERMA, A. (2001) A preconditioned conjugate gradient approach to linear equality constrained minimization. *Comput. Optim. Appl.*, **20**, 61–72.

CONN, A. R., GOULD, N. I. M. & TOINT, P. L. (2000) *Trust-Region Methods*. MPS-SIAM Series on Optimization. Philadelphia, PA: SIAM Publications.

DENNIS, J. E. & SCHNABEL, R. B. (1983) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall. Reprinted as Classics in Applied Mathematics, vol. 16. Philadelphia, PA: SIAM (1996).

DOLAN, E. D. & MORÉ, J. J. (2002) Benchmarking optimization software with performance profiles. *Math. Program. Ser. A*, **91**, 201–213.

DRUD, A. (1985) CONOPT: a GRG code for large sparse dynamic nonlinear optimization problems. *Math. Program.*, **31**, 153–191.

FACCHINEI, F. & LUCIDI, S. (1994) Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. *J. Optim. Theory Appl.*, **85**, 265–289.

FLETCHER, R. (1987) *Practical Methods of Optimization*, 2nd edn. Chichester, UK: John Wiley.

FLETCHER, R. & LEYFFER, S. (2002) Nonlinear programming without a penalty function. *Math. Program.*, **91**, 239–269.

FLETCHER, R. & SAINZ DE LA MAZA, E. (1989) Nonlinear programming and nonsmooth optimization by successive linear programming. *Math. Program.*, **43**, 235–256.

FRIEDLANDER, M. P., GOULD, N. I. M., LEYFFER, S. & MUNSON, T. S. (2007) A filter active-set trust-region method. *Technical Report Preprint ANL/MCS-P1456-097*. Argonne: Argonne National Laboratory.

GILBERT, J. C. (2007) SQPlab—A MATLAB software package for solving nonlinear optimization problems and optimal control problems. *Technical Report Version 0.4.1*. Rocquencourt: INRIA.

GILL, P. E., MURRAY, W. & SAUNDERS, M. A. (2002) SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM J. Optim.*, **12**, 979–1006.

GILL, P. E., MURRAY, W. & SAUNDERS, M. A. (2005) User's guide for SNOPT 7.1: a Fortran package for large-scale nonlinear programming. *Technical Report NA 05-2*. San Diego: Department of Mathematics, University of California.

GILL, P. E., MURRAY, W. & WRIGHT, M. H. (1981) *Practical Optimization*. London: Academic Press.

GOULD, N. I. M., HRIBAR, M. E. & NOCEDAL, J. (2001) On the solution of equality constrained quadratic problems arising in optimization. *SIAM J. Sci. Comput.*, **23**, 1375–1394.

GOULD, N. I. M., ORBAN, D. & TOINT, P. L. (2003) CUTEr and sifdec: a constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Softw.*, **29**, 373–394.

GOULD, N. I. M., ORBAN, D. & TOINT, P. L. (2005) Numerical methods for large-scale nonlinear optimization. *Acta Numerica*, **14**, 299–361.

GOULD, N. I. M. & ROBINSON, D. P. (2010a) A second derivative sqp method: global convergence. *SIAM J. Optim.*, **20**, 2023–2048.

GOULD, N. I. M. & ROBINSON, D. P. (2010b) A second derivative sqp method: local convergence and practical issues. *SIAM J. Optim.*, **20**, 2049–2079.

KELLER, C., GOULD, N. I. M. & WATHEN, A. J. (2000) Constraint preconditioning for indefinite linear systems. *SIAM J. Matrix Anal. Appl.*, **21**, 1300–1317.

NOCEDAL, J. & WRIGHT, S. J. (2006) *Numerical Optimization*, 2nd edn. Springer Series in Operations Research. New York: Springer.

OMOJOKUN, E. O. (1989) Trust region algorithms for optimization with nonlinear equality and inequality constraints. *Ph.D. Thesis*, University of Colorado, Boulder, CO, USA.

POWELL, M. J. D. (1978) A fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis, Dundee 1977*, vol. 630 (G. A. Watson ed.). Lecture Notes in Mathematics. Heidelberg, Berlin: Springer, pp. 144–157.

POWELL, M. J. D. (1983) Variable metric methods for constrained optimization. *Mathematical Programming: The State of the Art, Bonn 1982* (A. Bachem, M. Grötschel & B. Korte eds). Springer.

ROBINSON, S. M. (1974) Perturbed Kuhn–Tucker points and rates of convergence for a class of nonlinear programming algorithms. *Math. Program.*, **7**, 1–16.

SCHITTKOWSKI, K. (1981) The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian type line search function. *Numer. Math.*, **38**, 83–114.

VANDERBEI, R. J. & SHANNO, D. F. (1999) An interior point algorithm for nonconvex nonlinear programming. *Computat. Optim. Appl.*, **13**, 231–252.

WALTZ, R. A., MORALES, J. L., NOCEDAL, J. & ORBAN, D. (2006) An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math. Program. Ser. A*, **107**, 391–408.

Appendix: Test problems and numerical results

TABLE A1 *Number of iterations for SQP+ and C-SQP*

Name	*n*	Bounds	Ineq	Equal	SQP+	C-SQP
airport	84	168	42	0	11	89
biggsb1	144	286	0	0	72	188
bqpgasim	50	100	0	0	3	24
chenhark	200	200	0	0	4	413
clnlbeam	200	198	0	0	3	3
combustion	144	288	0	0	1	15
dixchlnv	100	100	0	50	28	−24
dnieper	57	112	0	24	23	−49
dual3	111	222	0	1	13	56
eg3	101	200	199	1	18	18
eigmaxa	101	200	0	101	6	6
eigmaxb	101	202	0	101	8	8
eigmina	101	202	0	101	2	2
eigminb	101	202	0	101	8	8
expfita	5	0	21	0	13	39
explin	144	288	0	0	97	112
explin2	144	288	0	0	72	95
grouping	100	200	0	125	1	1
haifas	7	0	9	0	10	10
haldmads	6	0	42	0	9	10
hanging	288	0	180	0	46	162
harkerp2	144	144	0	0	37	92
himmelbi	100	100	12	0	35	36
hs001	2	1	0	0	26	21
hs002	2	1	0	0	10	15
hs003	2	1	0	0	2	8
hs004	2	2	0	0	2	2
hs005	2	4	0	0	8	5
hs006	2	0	0	1	7	7
hs007	2	0	0	1	9	10
hs008	2	0	0	2	5	5
hs009	2	0	0	1	5	5
hs010	2	0	1	0	9	12
hs011	2	0	1	0	6	11
hs012	2	0	1	0	8	10
hs014	2	0	1	1	6	7
hs015	2	1	2	0	3	−7
hs018	2	4	2	0	6	8

A negative number indicates failure to converge.

TABLE A2 *Number of iterations for SQP+ and C-SQP*

Name	*n*	Bounds	Ineq	Equal	SQP+	C-SQP
hs019	2	4	2	0	6	6
hs020	2	2	3	0	8	−22
hs021	2	4	1	0	1	3
hs022	2	0	2	0	4	4
hs023	2	4	5	0	6	6
hs024	2	2	2	0	4	4
hs025	3	6	0	0	0	0
hs026	3	0	0	1	21	21
hs027	3	0	0	1	24	32
hs028	3	0	0	1	4	4
hs029	3	0	1	0	7	10
hs030	3	6	1	0	1	2
hs031	3	6	1	0	6	−10
hs032	3	3	1	1	3	4
hs033	3	4	2	0	4	4
hs034	3	6	2	0	7	7
hs035	3	3	1	0	3	9
hs036	3	6	1	0	2	−3
hs037	3	6	1	0	5	8
hs038	4	8	0	0	60	100
hs039	4	0	0	2	12	15
hs040	4	0	0	3	6	6
hs041	4	8	0	1	5	6
hs042	3	3	0	1	5	9
hs043	4	0	3	0	7	9
hs044	4	4	6	0	6	6
hs045	5	10	0	0	0	0
hs046	5	0	0	2	51	51
hs047	5	0	0	3	28	29
hs048	5	0	0	2	7	7
hs049	5	0	0	2	29	29
hs050	5	0	0	3	16	16
hs051	5	0	0	3	3	3
hs052	5	0	0	3	8	12
hs053	5	10	0	3	1	16
hs054	6	12	0	1	2	5
hs055	6	8	0	6	2	2
hs056	7	7	0	4	5	13
hs057	2	2	1	0	6	2
hs059	2	4	3	0	10	15
hs060	3	6	0	1	6	9
hs062	3	6	0	1	7	9
hs064	3	3	1	0	35	110
hs065	3	6	1	0	8	25
hs066	3	6	2	0	4	6

A negative number indicates failure to converge.

TABLE A3 *Number of iterations for SQP+ and C-SQP*

Name	*n*	Bounds	Ineq	Equal	SQP+	C-SQP
hs070	4	8	1	0	14	31
hs071	4	8	1	1	5	6
hs072	4	8	2	0	18	35
hs073	4	4	2	1	4	4
hs075	4	8	1	3	6	18
hs076	4	4	3	0	1	5
hs077	5	0	0	2	17	17
hs078	5	0	0	3	7	7
hs079	5	0	0	3	13	13
hs080	5	10	0	3	6	7
hs081	5	10	0	3	15	13
hs083	5	10	3	0	4	−6
hs085	5	10	36	0	0	0
hs086	5	5	6	0	5	9
hs087	9	18	0	4	67	88
hs089	3	0	1	0	23	60
hs090	4	0	1	0	24	25
hs091	5	0	1	0	18	39
hs092	6	0	1	0	26	38
hs093	6	6	2	0	76	31
hs095	6	12	4	0	2	2
hs096	6	12	4	0	2	2
hs097	6	12	4	0	13	13
hs098	6	12	4	0	13	13
hs100	7	0	4	0	8	16
hs100lnp	7	0	0	2	14	16
hs101	7	14	6	0	49	−77
hs102	7	14	6	0	32	−91
hs103	7	14	6	0	39	245
hs104	8	16	6	0	11	34
hs105	8	16	0	0	75	−94
hs106	8	16	6	0	24	46
hs108	9	1	13	0	11	12
hs110	10	20	0	0	6	7
hs111	10	20	0	3	10	47
hs111lnp	10	0	0	3	48	47
hs113	10	0	8	0	5	17
hs116	13	26	15	0	348	95
hs117	15	15	5	0	17	−17
hs118	15	30	17	0	7	13
hs119	16	32	0	8	4	−19
hs268	5	0	5	0	2	32
hs3mod	2	1	0	0	5	7
hs44new	4	4	5	0	5	5

A negative number indicates failure to converge.

TABLE A4 *Number of iterations for SQP+ and C-SQP*

Name	*n*	Bounds	Ineq	Equal	SQP+	C-SQP
jnlbrng1	144	144	0	0	7	36
jnlbrng2	144	144	0	0	4	45
jnlbrnga	144	144	0	0	3	31
jnlbrngb	144	144	0	0	1	61
loadbal	31	42	20	11	17	79
makela3	21	0	20	0	22	−19
mccormck	144	288	0	0	4	17
minsurfo	144	144	0	0	5	43
ncvxbqp1	144	288	0	0	2	−3
ncvxbqp2	144	288	0	0	4	−6
ncvxbqp3	144	288	0	0	124	142
nobndtor	144	144	0	0	5	37
nonscomp	144	288	0	0	9	59
obstclae	169	338	0	0	11	28
obstclbm	169	338	0	0	6	26
optctrl3	118	0	1	79	7	−206
optctrl6	118	0	1	79	7	−206
optprloc	30	60	29	0	6	12
pentdi	144	144	0	0	1	2
probpenl	144	288	0	0	2	2
prodpl0	60	60	9	20	8	8
prodpl1	60	60	9	20	6	6
qrtquad	144	20	0	0	36	57
qudlin	144	288	0	0	2	−3
rk23	17	6	0	11	9	9
s368	144	288	0	0	0	0
synthes1	6	12	6	0	5	8
torsion1	144	288	0	0	4	17
torsion2	144	288	0	0	5	20
torsion3	144	288	0	0	2	9
torsion4	144	288	0	0	5	13
torsion5	144	288	0	0	1	3
torsion6	144	288	0	0	3	11
torsiona	144	288	0	0	4	17
torsionb	144	288	0	0	8	18
torsionc	144	288	0	0	2	8
torsiond	144	288	0	0	4	17
torsione	144	288	0	0	1	3
torsionf	144	288	0	0	3	8
trimloss	142	264	52	20	183	114

A negative number indicates failure to converge.