Technical Research Report

Systems Research Center

Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
Industry and the University

Avoiding the Maratos Effect by Means of a Nonmonotone Line Search II. Inequality Constrained Problems - Feasible Iterates

by J.F. Bonnans, E.R. Panier, A.L. Tits and J. Zhou

TR 89-42r1

SIAM J. NUM. ANAL.
Vol. x, No. x, pp. x–xx, xxx 19xx

000

high
end

AVOIDING THE MARATOS EFFECT BY MEANS OF A NONMONOTONE LINE SEARCH II. INEQUALITY CONSTRAINED PROBLEMS – FEASIBLE ITERATES*

J. FRÉDÉRIC BONNANS†, ELIANE R. PANIER‡, ANDRÉ L. TITS$ AND JIAN ZHOU$

Abstract. When solving inequality constrained optimization problems via Sequential Quadratic Programming (SQP), it is potentially advantageous to generate iterates that all satisfy the constraints: all quadratic programs encountered are then feasible and there is no need for a surrogate merit function. (Feasibility of the successive iterates is in fact required in many contexts such as in real-time applications or when the objective function is not defined outside the feasible set.) It has recently been shown that this is indeed possible, by means of a suitable perturbation of the original SQP iteration, without losing superlinear convergence. In this context, the well known Maratos effect is compounded by the possible infeasibility of the full step of one even close to a solution. These difficulties have been accommodated by making use of a suitable modification of a "bending" technique proposed by Mayne and Polak, requiring evaluation of the constraints function at an auxiliary point at each iteration.

In part I of this two-part paper, it was shown that, when feasibility of the successive iterates is not required, the Maratos effect can be avoided by combining Mayne and Polak's technique with a nonmonotone line search proposed by Grippo, Lampariello and Lucidi in the context of unconstrained optimization, in such a way that, except possibly at a few early iterations, function evaluations are no longer performed at auxiliary points. In this second part, we show that feasibility can be restored without resorting to additional constraint evaluations, by adaptively estimating a bound on the second derivatives of the active constraints. Extension to constrained minimax problems is briefly discussed.

Key words. constrained optimization, sequential quadratic programming, Maratos effect, superlinear convergence, feasibility

AMS(MOS) subject classifications. 90C30, 65K10

1. Introduction.

Consider the optimization problem

$$(P) \qquad \min f(x) \quad \text{s.t. } g(x) \le 0$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions. It has been shown that a suitable modification of the Sequential Quadratic Programming (SQP) iteration can be used to solve efficiently such problems while generating iterates that all satisfy the constraints [13], [15]. Preserving feasibility is an important attribute in many contexts, e.g., (i) when the objective function is not defined outside the feasible set or

*Received by the editors xxx xx, 19xx; accepted for publication (in revised form) xxx xx, 19xx. This research was supported in part by NSF's Engineering Research Centers Program No. NSFD-CDR-88-03012 and by NSF grants No. DMC-84-51515 and DMC-88-15966. The third author performed parts of this work while visiting INRIA (project META2), Rocquencourt, France, in the fall of 1988.

†INRIA, Domaine de Voluceau, BP 105, Rocquencourt, 78153 Le Chesnay Cedex, France.

‡Chase Manhattan Bank. Was with the Systems Research Center, University of Maryland, College Park, MD 20742.

$Systems Research Center and Electrical Engineering Department, University of Maryland, College Park, MD 20742.

(ii) in real-time applications, when it is crucial that a feasible solution be available at the next "stopping time". From a computational point of view, when this property is satisfied, advantages are that the quadratic programs successively constructed all have a non empty feasible set and that the objective function itself can be used as a merit function in the line search. On the other hand the condition that the line search yield a feasible next iterate renders increasingly difficult the question of devising a mechanism to ensure that the full step of one is eventually taken, an imperative requirement if superlinear convergence is to take place. (The possible undesirable truncation of the step due to failure of the merit function to decrease when a full step is taken was first pointed out by Maratos [11].) In particular the watchdog technique [1], by which the full step of one is tentatively accepted if sufficient decrease was obtained in recent iterations, is of no help here. In [13], [15], the issue is resolved by making use of a "bending" technique employed by Mayne and Polak [12], suitably adapted for restoring feasibility. The appropriate amount of bending is determined via evaluation of the constraints at an auxiliary point at each iteration.

In part I of this two-part paper [14], it was shown that, when feasibility of the successive iterates is not required, the "Maratos effect" can be avoided by combining Mayne and Polak's technique with a nonmonotone line search proposed by Grippo, Lampariello and Lucidi in the context of unconstrained optimization, in such a way that, except possibly at a few early iterations, constraint evaluations are no longer performed at auxiliary points. In this second part, we show that feasibility can be restored without resorting to additional constraint evaluations, by adaptively estimating a bound on the second derivatives of the active constraints.

Based on ideas similar to those used in part I, one can show that, if a sequence of iterates $\{x_k\}$ is generated by the basic SQP iteration for (P), i.e., $x_{k+1} = x_k + d_k^0$ where d_k^0 solves the quadratic program $QP(x_k, H_k)$

$$\begin{aligned} \text{min}_d \quad & \frac{1}{2}\langle d, H_k d\rangle + \langle \nabla f(x_k), d\rangle \\ \text{s.t.} \quad & g_j(x_k) + \langle \nabla g_j(x_k), d\rangle \le 0 \quad j = 1, \ldots, m \ , \end{aligned} \tag{1.1}$$

with H_k a suitable uniformly symmetric positive definite estimate to the Hessian of the Lagrangian; then, if x_0 is sufficiently close to a strong local minimizer x^* for (P) and if the entire sequence $\{x_k\}$ happens to be feasible,

$$f(x_{k+1}) \le f(x_{k-3}) + \alpha\langle \nabla f(x_k), x_{k+1} - x_k\rangle \tag{1.2}$$

will hold for k large enough, with α any positive number. Thus, a line search rule requiring that the condition

$$f(x_k + td_k^0) \le \max_{\ell=0,\ldots,3} f(x_{k-\ell}) + \alpha t\langle \nabla f(x_k), d_k^0\rangle \ ,$$

be satisfied would eventually *always* accept the full step of one, provided it does so three times in a row. Initialization of this process could be performed, as done in part I, by making use of Mayne and Polak's correction. Again, locally around x^*, evaluation of the constraints at auxiliary points would not be necessary. Global convergence of a corresponding scheme was proved in part I based on a result in [5].[1] In the present case however, a major difficulty remains: the full step of one will likely be rejected due to infeasibility.

[1] The fact that d_k^0 is a direction of descent for f is crucial.

It turns out (see Theorem 3.8 below) that an inequality analogous to (1.2), namely

$$f(x_{k+1}) \leq f(x_{k-3}) + \alpha\langle \nabla f(x_k), d_k^0 \rangle \ ,$$

still holds if the sequence $\{x_k\}$ is constructed via the iteration $x_{k+1} = x_k + d_k$, where d_k satisfies[2]

$$d_k = d_k^0 + O(\|d_k^0\|^2) \ . \tag{1.3}$$

In [13] and [15], SQP-type algorithms are proposed that make use of a *feasible* direction d_k that satisfies (1.3). Unfortunately to achieve feasibility of the full step of one close to a solution, they resort to a correction of the Mayne-Polak type that require evaluation of the constraints at auxiliary points, which is precisely what we aim to avoid here. The question to investigate is thus whether restoration of mere feasibility (rather than feasibility and descent on f, as in [13] and [15]) for the full step of one can be achieved without such additional constraint evaluation. To this end, consider a "local" search direction d_k^ℓ such that

$$d_k^\ell = d_k^0 + O(\|d_k^0\|^2) \tag{1.4}$$

satisfying

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^\ell \rangle \leq -C\|d_k^0\|^2, \quad j = 1, \ldots, m \ , \tag{1.5}$$

for given $C > 0$. Under mild assumptions, when x_k is close to x^*, d_k^0 is small and, if the gradients of the active constraints at x^* are linearly independent, it is always possible to construct such d_k^ℓ. If (1.5) holds, the sequence $\{x_k\}$ constructed by the iteration $x_{k+1} = x_k + d_k^\ell$ will satisfy

$$\begin{aligned} g_j(x_{k+1}) &= g_j(x_k) + \langle \nabla g_j(x_k), d_k^\ell \rangle + \frac{1}{2}\langle d_k^\ell, \frac{\partial^2 g_j}{\partial x^2}(x_k + \xi_{j,k} d_k^\ell) d_k^\ell \rangle \\ &\leq -C\|d_k^0\|^2 + \frac{1}{2}\|d_k^\ell\|^2 \|\frac{\partial^2 g_j}{\partial x^2}(x_k + \xi_{j,k} d_k^\ell)\|, \ j = 1, \ldots, m \end{aligned}$$

for some $\xi_{j,k} \in [0,1]$. And if (1.4) holds, for k large enough we will have $g_j(x_{k+1}) \leq 0$, $j = 1, \ldots, m$, provided $2C$ is strictly larger than the largest among all eigenvalues of the Hessians $\frac{\partial^2 g_j}{\partial x^2}(x^*)$, $j = 1, \ldots, m$. While these Hessians are obviously unknown, one could attempt to adaptively obtain a suitably large value of C, by increasing C whenever the step of one along d_k^ℓ is not feasible. It will be shown below that this can indeed be achieved.

A final difficulty stems from the fact that, away from x^*, d_k^ℓ may not be a descent direction for f. Indeed, conditions (1.4) and (1.5) and the descent property are usually incompatible. This can be addressed by resorting, whenever the full step of one along d_k^ℓ is not accepted for the descent criterion, to a search along the arc,

$$x(t) = x_k + t d_k^g + t^2 \tilde{d}_k$$

---

[2]Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$ as well as the corresponding operator norm in $\mathbb{R}^{n \times n}$.

with d_k^g a "global" feasible descent direction and $\tilde{d}_k$ a correction of the Mayne-Polak type, as in [13] and [15]. Borrowing from the scheme used in [15], both d_k^ℓ and d_k^g will be chosen as convex combination of d_k^0 and a direction $d^1(x_k)$ obtained via a certain map $d^1(\cdot)$. In [15], $d^1(x)$ is essentially any feasible descent direction at x and thus vanishes at any Karush-Kuhn-Tucker (KKT) point. In this context however, to ensure that (1.5) can be achieved close to x^*, we will require that $d^1(x)$ be a (nonzero) direction of strict feasibility even at KKT points. Clearly then, one cannot any more require that $d^1(x)$ be a direction of descent for f.

The balance of this paper is organized as follows. In §2, we propose an algorithm based on the foregoing ideas. A convergence analysis is carried out in §3. In §4, extension to constrained minimax problems is briefly discussed. Implementation issues and numerical experiments are discussed in §5. Finally, §6 is devoted to concluding remarks. To avoid a loss of continuity, several proofs are given in an appendix.

2. The algorithm. A point x^* is said to be a *Karush-Kuhn-Tucker (KKT) point* for (P) if $g(x^*) \leq 0$ and there exist multipliers μ_j^*, $j = 1, \ldots, m$, with $\mu_j^* \geq 0$, such that

$$\nabla_x L(x^*, \mu^*) = 0$$

and

$$\mu_j^* g_j(x^*) = 0, \ j = 1, \ldots, m$$

where $L(x, \mu)$ denotes the Lagrangian function

$$L(x, \mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x).$$

In the sequel, we denote by X the feasible set for (P), i.e.,

$$X = \{x \mid g(x) \leq 0\}.$$

We make the following assumptions.

A1. The functions f, g_j, $j = 1, \ldots, m$ are continuously differentiable.

A2. The set $\{x \in X \mid f(x) \leq f(x_0)\}$ is compact.

A3. $\forall x \in X$, $\{\nabla g_j(x),\ j \in I(x)\}$ are linearly independent,

where

$$I(x) = \{j \mid g_j(x) = 0\} .$$

We also assume that there exist $\sigma_1, \sigma_2 > 0$ such that

$$\sigma_1 \|x\|^2 \leq \langle x, H_k x \rangle \leq \sigma_2 \|x\|^2, \quad \forall x \in \mathbb{R}^n, \quad \forall k \in \mathbb{N}. \tag{2.1}$$

As indicated above, Algorithm 2.1 below makes use of two search directions, d_k^ℓ and d_k^g, both of which are convex combinations of d_k^0 and of a feasible direction $d^1(x_k)$ obtained via a map $d^1(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$. This map is required to be continuous[3] and to yield for every $x \in X$ (including KKT points) a direction $d^1(x)$ satisfying

$$g_j(x) + \langle \nabla g_j(x), d^1(x) \rangle < 0, \quad j = 1, \ldots, m . \tag{2.2}$$

[3]For simplicity of exposition; the results still hold with milder assumptions.

In view of Assumption A3, such a direction can, for example, be obtained as the solution of

$$\min_d \frac{1}{2}\|d\|^2 + \max_{j=1,\ldots,m}\{g_j(x) + \langle \nabla g_j(x), d\rangle\} \ .$$

Further clarification is in order concerning the specific construction of d_k^ℓ and d_k^g. The "local" search direction d_k^ℓ is constructed based on a constant C_k, corresponding to C in (1.5), which is iteratively adapted as suggested in the introduction. Essentially, it is increased if $\|d_k^0\|$ is reasonably small (indicating that x^* is nearby) but $x_k + d_k^\ell$ is not feasible. If $x_k + d_k^\ell$ is feasible, C_k is kept to its previous value and if $\|d_k^0\|$ is large, it is reset to some small value. Next, it is easily checked that if C_k remains bounded as $k \to \infty$ (which will be shown to be true) and d_k^ℓ is constructed according to

$$d_k^\ell = (1-\rho_k^\ell)d_k^0 + \rho_k^\ell d^1(x_k),$$

with $\rho_k^\ell \in [0,1]$ as small as possible subject to satisfaction of (1.5), the requirement (1.4) will be satisfied. Away from x^* however it may be impossible to satisfy (1.5) with $\rho_k^\ell \in [0,1]$. A suitable choice in such case would be $\rho_k^\ell = 1$. In Step 1 *ii* in Algorithm 2.1 below, ρ_k^ℓ is constructed essentially according to these rules, with the additional feature that ρ_k^ℓ (and thus d_k^ℓ) is forced to go to zero whenever d_k^0 does, to preserve global convergence even if C_k tends to grow without bound. For the "global" direction d_k^g, the requirement is that it should be a feasible descent direction, and that

$$d_k^g = d_k^0 + O(\|d_k^0\|^2) \ .$$

This can be achieved by selecting

$$d_k^g = (1-\rho_k^g)d_k^0 + \rho_k^g d^1(x_k)$$

with $\rho_k^g \in [0, \rho_k^\ell]$, as large as possible subject to the condition

$$\langle \nabla f(x_k), d_k^g\rangle \le \theta\langle \nabla f(x_k), d_k^0\rangle \ ,$$

where $\theta \in (0,1)$ is a fixed parameter. This is done in Step 1 *v* below.

2.1. Algorithm.
Parameters. $\alpha \in (0,1/2)$, $\beta \in (0,1)$, $\theta \in (0,1)$, $\gamma \in (2,3)$, $\underline{C}$, $\underline{d} > 0$.
Data. $x_0 \in X$, $H_0 \in \mathbb{R}^{n\times n}$, symmetric positive definite, $C_0 = \underline{C}$.
Step 0. Initialization. Set $k = 0$.
Step 1. Computation of a new iterate.
i. Compute d_k^0 solution of the quadratic program $QP(x_k, H_k)$.
If $d_k^0 = 0$ stop.
ii. Compute $d_k^1 = d^1(x_k)$, let $v_k = \min\{C_k\|d_k^0\|^2, \|d_k^0\|\}$ and define values $\rho_{k,j}$ for $j = 1, \ldots, m$ by $\rho_{k,j}$ equal to zero if

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^0\rangle \le -v_k$$

or equal the maximum ρ in $[0,1]$ such that

$$g_j(x_k) + \langle \nabla g_j(x_k), (1-\rho)d_k^0 + \rho d_k^1\rangle \ge -v_k$$

otherwise. Finally, let $\rho_k^\ell = \max_{j=1,\ldots,m}\{\rho_{k,j}\}$.

iii. Obtain a "local" direction

$$d_k^\ell = (1-\rho_k^\ell)d_k^0 + \rho_k^\ell d_k^1.$$

iv. If

$$f(x_k + d_k^\ell) \le \max_{\ell=0,\dots,3}\{f(x_{k-\ell})\} + \alpha\langle\nabla f(x_k), d_k^0\rangle$$

and

$$g_j(x_k + d_k^\ell) \le 0, \quad j = 1,\dots,m,$$

set $t_k = 1$, $x_{k+1} = x_k + d_k^\ell$ and go to Step 2. Otherwise, go to Step 1 *v*.

v. Obtain a "global" direction

$$d_k^g = (1-\rho_k^g)d_k^0 + \rho_k^g d_k^1 \ ,$$

where ρ_k^g is the largest number in $[0, \rho_k^\ell]$ such that

$$\langle\nabla f(x_k), d_k^g\rangle \le \theta\langle\nabla f(x_k), d_k^0\rangle.$$

vi. Compute a "correction" $\tilde{d}_k$ by solving the quadratic program $\widetilde{QP}(x_k, d_k^g, H_k)$

$$\begin{aligned} &\min_{\tilde{d}} \quad \frac{1}{2}\langle(d_k^g + \tilde{d}), H_k(d_k^g + \tilde{d})\rangle + \langle\nabla f(x_k), \tilde{d}\rangle \\ &\text{s.t.} \quad g_j(x_k + d_k^g) + \langle\nabla g_j(x_k), \tilde{d}\rangle \le -\|d_k^g\|^\gamma, \quad j = 1,\dots,m. \end{aligned}$$

If there is no solution or if $\|\tilde{d}_k\| > \|d_k^0\|$, set $\tilde{d}_k = 0$.

vii. Compute t_k, the first number t in the sequence $\{1, \beta, \beta^2, \cdots\}$ satisfying

$$f(x_k + td_k^g + t^2\tilde{d}_k) \le \max_{\ell=0,\dots,3}\{f(x_{k-\ell})\} + \alpha t\langle\nabla f(x_k), d_k^g\rangle,$$

$$g_j(x_k + td_k^g + t^2\tilde{d}_k) \le 0, \quad j = 1,\dots,m$$

and set $x_{k+1} = x_k + t_k d_k^g + t_k^2\tilde{d}_k$.

Step 2. Updates.

· Compute a new symmetric positive definite approximation H_{k+1} to the Hessian of the Lagrangian.

· If $\|d_k^0\| > \underline{d}$, set $C_k = \underline{C}$. Otherwise, if $g_j(x_k + d_k^\ell) \le 0, \quad j = 1,\dots,m$, set $C_{k+1} = C_k$. Otherwise, set $C_{k+1} = 2C_k$.

· Increase k by 1.

· Go back to Step 1. □

Note that while Algorithm 2.1 involves possible auxiliary constraint evaluations at $x_k + d_k^\ell$ and $x_k + d_k^g$ (Steps 1 *iv* and 1 *vi*) it can be easily modified so as to require at most one auxiliary constraint evaluation per iteration (either at $x_k + d_k^\ell$ or at $x_k + d_k^g$). See §5 below.

3. Convergence results.

3.1. Global convergence.

In view of the feasibility of the iterates and the positive definiteness of H_k (see (2.1)), the quadratic program $QP(x_k, H_k)$ always has a unique solution d_k^0. Using the optimality conditions associated with $QP(x_k, H_k)$ it can be checked that d_k^0 is equal to zero if and only if x_k is a KKT point for (P) and that, if d_k^0 is not zero then

$$\langle \nabla f(x_k), d_k^0 \rangle \leq -\langle d_k^0, H_k d_k^0 \rangle + \sum_{j=1}^{m} \mu_{k,j} g_j(x_k) \tag{3.1}$$

where μ_k denotes the nonnegative multiplier vector associated with $QP(x_k, H_k)$. Therefore, in view of the feasibility of the iterates, the positive definiteness of H_k (see (2.1)) and the definition of d_k^g, it follows that

$$\langle \nabla f(x_k), d_k^g \rangle < 0$$

and

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^g \rangle < 0, \; j = 1, \dots, m$$

so that, using a standard argument, it can be proven that, whenever a line search is performed in Step 1 *vii*, it yields a step $t_k = \beta^{j(k)}$ for some finite $j(k) \in \mathbb{N}$. The algorithm is thus well defined. In the sequel, we assume that stop at Step 1 *i* never occurs so that a complete sequence $\{x_k\}$ is generated. We proceed to show that every accumulation point of $\{x_k\}$ is a KKT point for (P). The following results are proven in the appendix.

PROPOSITION 3.1. *The sequence $\{x_k\}$ is bounded and the sequences $\{t_k d_k^0\}$ and $\{||x_{k+1} - x_k||\}$ both converge to zero.*

THEOREM 3.2. *Let x^* be an accumulation point of the sequence generated by the algorithm and let $\{x_k\}_{k \in K}$ be any subsequence converging to x^*. Then, x^* is a KKT point of (P) and the subsequence $\{d_k^0\}_{k \in K}$ converges to zero.*

3.2. Superlinear convergence. In order to prove superlinear convergence, we assume some more regularity on the functions involved. Assumption A1 is replaced by

A1′. The functions f, g_j, $j = 1, \dots, m$ are three times continuously differentiable. Let x^* be an accumulation point of the sequence generated by the algorithm (a KKT point of (P) in view of Theorem 3.2). The KKT multiplier at x^* is denoted by μ^*. We further assume that the *second order sufficiency conditions with strict complementary slackness* are satisfied at x^*, i.e., that $\mu_j^* > 0 \; \forall j \in I(x^*)$ and that the Hessian of the Lagrangian function $\nabla_{xx} L(x^*, \mu^*)$ is positive definite on the subspace

$$\{p | < \nabla g_j(x^*), p >= 0 \quad \forall j \in I(x^*)\} \; .$$

Many of the proofs of the results stated below are very similar to the ones in Part I of this two-part paper and thus are omitted.

PROPOSITION 3.3. *The entire sequence $\{x_k\}$ converges to x^*.*

Proof. The proof is similar to the one of Proposition 3.4 in [14]. □

PROPOSITION 3.4. *There exists $\bar{C} > 0$ such that $C_k \leq \bar{C}$, $\forall k$.*

Proof. In view of Proposition 3.3, the properties of $d^1(\cdot)$ and the definition of d_k^l in Step 1 *iii*, $\exists \; M > 0$ such that

$$||d_k^\ell|| \leq M ||d_k^0|| \quad \forall k. \tag{3.2}$$

Let

$$N = \frac{1}{2} \max_{j \in \{1,\dots,m\}, \; t \in (0,1), k \in \mathbb{N}} \{ || \frac{\partial^2 g_j}{\partial x^2}(x_k + t d_k^\ell) || \}$$

and let $C = MN$. In view of the regularity of the functions g_j's (Assumption A1′) and the boundedness of $\{x_k\}$ and $\{d_k^\ell\}$, this quantity is well defined. For any $j \in \{1, \ldots, m\}$, we have

$$g_j(x_k + d_k^\ell) \le g_j(x_k) + \langle \nabla g_j(x_k), d_k^\ell \rangle + N \| d_k^\ell \|^2.$$

In view of the definition of d_k^ℓ in Step 1 *iii*, the properties of $d^1(\cdot)$ and the fact that $\{d_k^0\}$ converges to zero, for k large enough we get

$$g_j(x_k + d_k^\ell) \le -\min\{C, C_k\} \| d_k^0 \|^2 + N \| d_k^\ell \|^2.$$

Therefore, from (3.2)

$$g_j(x_k + d_k^\ell) \le -\min\{C, C_k\} \| d_k^0 \|^2 + MN \| d_k^0 \|^2.$$

This completes the proof since, if C_k keeps increasing, for k big enough $C_k \ge C$ and, from the definition of C we obtain

$$g_j(x_k + d_k^\ell) \le 0$$

and, in view of Step 2 in the algorithm, C_k would remain constant, a contradiction. $\square$

Assume now that the approximations H_k to the Hessian of the Lagrangian at x^* satisfy a property of the type

$$\frac{\| P_k (H_k - \nabla_{xx}^2 L(x^*, \mu^*)) P_k d_k^0 \|}{\| d_k^0 \|} \to 0$$

as k goes to infinity, where the matrices P_k are defined by

$$P_k = I - R_k (R_k^T R_k)^{-1} R_k^T$$

with $R_k = [\nabla g_j(x_k) \mid j \in I(x^*)] \in \mathbb{R}^{n \times |I(x^*)|}$. (Note that, in view of the independence of the gradients of the active constraints at x^* (Assumption A3), the matrices $R_k^T R_k$ are invertible for k large enough.) We proceed to show that a step of one is always accepted for k large enough and that two-step superlinear convergence occurs. The next proposition states that, close to x^*, the active constraints at x^* are correctly identified by $QP(x_k, H_k)$.

PROPOSITION 3.5. *For k large enough,*

i. the multipliers μ_k^0 associated with the solutions d_k^0 of $QP(x_k, H_k)$ satisfy

$$\{\mu_k^0\} \to \mu^*$$

and

$$\mu_{k,j}^0 = 0, \quad \forall j \notin I(x^*).$$

ii.

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle = 0, \quad \forall j \in I(x^*).$$

iii.

$$d_k^\ell = d_k^0 + O(\| d_k^0 \|^2).$$

and

$$d_k^g = d_k^0 + O(\|d_k^0\|^2).$$

Proof. The proof of the first two statements is very similar to that of Proposition 4.2 in [13]. *iii* follows from Theorem 3.2, Propositions 3.3 and 3.4, the definition of d_k^ℓ and d_k^g in Step 1 *iii* and *v* and the properties of $d^1(\cdot)$. □

The following property states that eventually a step of one is accepted.

PROPOSITION 3.6. *For k large enough, (i)* $\widetilde{QP}(x_k, d_k^g, H_k)$ *always admits a solution* $\tilde{d}_k$ *which satisfies*

$$\tilde{d}_k = O(\|d_k^0\|^2)$$

and (ii) a step of one is accepted whenever the line search is performed.

Proof. Similarly to what is done in the proof of Proposition 3.6 in Part I of this two-part paper [14], it can be proven that (i) holds. Also, making use of Proposition 3.5 and observing that $d_k = P_k d_k + \hat{d}_k$ with $\|\hat{d}_k\| = O(\|g(x_k)\|)$ (see proof of Lemma 4.5 in [13]) it can be shown that

$$f(x_k + d_k^g + \tilde{d}_k) \leq \max_{\ell=0,\dots,3}\{f(x_{k-\ell})\} + \alpha\langle\nabla f(x_k), d_k^g\rangle.$$

Finally, expanding $g_j(x_k+d_k^g+\tilde{d}_k)$ about $x_k+d_k^g$, using the properties of $\tilde{d}_k$, Proposition 3.5 and the fact that $\gamma \in (2,3)$, it can be shown that, for k large enough,

$$g_j(x_k + d_k^g + \tilde{d}_k) \leq 0, \quad j = 1,\dots,m.$$

□

Finally, the convergence rate properties of SQP-type methods is preserved and auxiliary constraint evaluations are performed in the early iterations only.

THEOREM 3.7. *Under the stated assumptions, the convergence is two-step superlinear, i.e.,*

$$\lim_{k\to\infty}\frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.$$

Moreover,

$$\|x_{k+1} - x^*\| = O(\|x_k - x^*\|)\ .$$

□

THEOREM 3.8. *For k large enough,* d_k^ℓ *is always used and a correction* $\tilde{d}_k$ *is not computed.* □

This is the key result. Its proof is given in the appendix.

4. A simple extension: constrained minimax problems. It is well known that problems of the type

$$\text{(4.1)} \qquad \begin{array}{ll} \text{minimize} & \max\{f_1(x),\dots,f_p(x)\} \\ & \text{s.t. } g_j(x) \leq 0, \quad j = 1,\dots,m \end{array}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1,\dots,p$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1,\dots,m$ are smooth functions can be reformulated as a standard smooth constrained problem by introducing an auxiliary variable. Simple modifications of the SQP iteration that eliminate this auxiliary variable have been suggested [7], [8] and it has been shown that SQP-type algorithms generating feasible iterates can be adapted to include such modifications [16].

Algorithm 2.1 can similarly be extended to problem (4.1) in a straightforward manner. Specifically, the function $f : \mathbb{R}^n \to \mathbb{R}$ must now be taken as

$$f(x) = \max_{i=1,\ldots,p} f_i(x) \ ,$$

expressions of the form $\langle \nabla f(x), d \rangle$ in Step 1 *iv*, 1 *v* and 1 *vii* must be replaced by corresponding first variations $f'(x, d)$ with $f' : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ given by

$$f'(x, d) = \max_{i=1,\ldots,p} \{f_i(x) + \langle \nabla f_i(x), d \rangle\} - f(x).$$

In Step 1 *vi*, $\langle \nabla f(x_k), \tilde{d} \rangle$ must be replaced by

$$\tilde{f}'(x_k + d_k^0, x_k, \tilde{d}) = \max_{i=1,\ldots,p} \{f_i(x_k + d_k^0) + \langle \nabla f_i(x_k), \tilde{d} \rangle\} - f(x_k + d_k^0).$$

Finally, matrices H_k must now approximate the Hessian of the corresponding Lagrangian

$$L(x, \lambda, \mu) = \sum_{i=1}^{p} \lambda_i f_i(x) + \sum_{i=1}^{m} \mu_i g_i(x) \ .$$

5. Numerical experiments. An efficient implementation of the algorithm described in this paper has been completed (FSQP Version 2.0 [19]). In this implementation, given x, the direction $d^1(x)$ is defined to be the value of d^1 at which

$$\min_d \frac{\eta}{2} ||d^1||^2 + \max_{j=1,\ldots,m} \{g_j(x) + \langle \nabla g_j(x), d \rangle\}$$

is achieved, with $\eta = 3.0$. The other parameter values are $\alpha = 10^{-7}$, $\beta = 0.5$, $\theta = 0.2$, $\gamma = 2.5$, $\underline{C} = 0.01$, and $\underline{d} = 5.0$. The initial Hessian approximation H_0 is taken to be the identity. H_k is updated by means of the BFGS formula with Powell's modification [17]. The following minor modifications with respect to the algorithm as described in §2 were found to be beneficial and were implemented:

(*i*) Step 1 *v* is performed before Steps 1 *iii* and 1 *iv* and, in Step 1 *iii*, ρ^g is used instead of ρ_k^ℓ if the step of one was not accepted at the previous iteration or if $\rho_k^\ell > 0.5$ (this reduces the number of auxiliary constraint evaluations, in the spirit of the suggestion made at the very end of §2). Also, if $m = 0$ (in particular, in the case of unconstrained minimax problems), Steps 1 *ii*, 1 *iii* and 1 *v* are skipped and d_k^ℓ and d_k^g are set to d_k^0.

(*ii*) In the computation of $\tilde{d}$, $||d_k^g||^\gamma$ is replaced by $min\{0.01||d_k^g||, ||d_k^g||^\gamma\}$, to prevent $\tilde{d}$ from being too large in the early iterations.

(*iii*) In Step 2, if $||d_k^0|| > \underline{d}$, C_{k+1} is set to $max\{0.5\ C_k, \underline{C}\}$ to prevent too rapid a decrease of C_k. If $||d_k^0|| \geq \underline{d}$ and $g_j(x_k + d_k^\ell) > 0$ for some $j \in \{1, \ldots, m\}$, C_{k+1} is set to 10 C_k instead of 2 C_k.

(*iv*) The shopping criterion in Step 1 *i* is unsuitable for implementation. Instead, in FSQP, execution is terminated if the gradient of the Lagrangian at the current point, with the multipliers obtained in solving $QP(x_k, H_k)$, is less than some specified $\epsilon > 0$.

The FSQP code includes special provisions for efficient handling of affine constraints and it also accepts affine *equality* constraints. Such extensions are straightforward (see [19] for details). The extension to constrained minimax problems suggested in §4 is also implemented.

Results on two sets of experiments are presented in Tables 1 and 2. All computations were performed on a Sun 4/SPARCstation 1. For the first set of problems, gradients were computed analytically; for the second set, they were computed by finite differences (for the ith component, the perturbation parameter was $10^{-7} max\{1, |x_k^i|\}$).

Table 1 contains results obtained on test problems from [9]. The new algorithm (FSQP-NL) is compared to the algorithm analyzed in [15] (FSQP-AL), to the authors knowledge the best available "feasible iterate" algorithm.† It is observed that on all problems the number of nonlinear constraint evaluations is lower in FSQP-NL, often dramatically so. No significant difference is observed in the number of objective function evaluations.

Results obtained on selected minimax problems are summarized in Table 2. Problems BARD, DAVD2, F&R, HETTICH, and WATS are from [18]; CB2, CB3, R-S, WONG and COLV are from [2; Examples 5.1-5]; MAD1 to MAD8 are from [10, Examples 1-8]. Some of these test problems allow one to freely select the number of variables; problems WATS-6 and WATS-20 correspond to 6 and 20 variables, respectively, and MAD8-10, MAD8-30 and MAD8-50 to 10, 30 and 50 variables respectively. Problems BARD down to MAD8 are unconstrained or linearly constrained minimax problems. Unable to find nonlinearly constrained minimax test problems in the literature, we constructed problems P43M through P117M from problems 43, 84, 113 and 117 in [9] by removing certain constraints and including instead additional objectives of the form $f_i(x) = f(x) + \alpha_i g_j(x)$ where the α_i's are positive scalars and $g_j(x) \leq 0$. Specifically, P43M is constructed from problem 43 by taking out the first two constraints and including two corresponding objectives with $\alpha_i = 15$ for both; P84M similarly corresponds to problem 84 without constraints 5 and 6 but with two corresponding additional objectives, with $\alpha_i = 20$ for both; for P113M the first three linear constraints from problem 113 were turned into objectives, with $\alpha_i = 10$ for all three; for P117M, the first two nonlinear constraints were turned into objectives, again with $\alpha_i = 10$ for both. In Table 2, the performance of FSQP is also compared with that of the algorithms proposed in [3] (NM) and [10] (MS). To make such comparison meaningful, we attempted to best approximate the stopping rule used in each of the references. Thus the stopping criterion given in *(iv)* above was not used, but rather (i) for problems BARD down to WONG, execution was terminated when $||d_k^0||$ was smaller than the corresponding value of ϵ in the EPS column (this was also done for problems P43M down to P117M), and (ii) for problems MAD1 down to MAD8-50, execution was terminated when $||d_k^0||$ was smaller than $||x_k||$ times the corresponding value of ϵ in the EPS column. The other columns are as in Table 1 (see bottom of that table), except that NOBJ gives the number of objective functions (in the max), NMF the total number of evaluations of the *set* of objective functions (rather than the total number of evaluations of scalar objective functions, to allow better comparison with the results given in the references), and OBJMAX the final value of the max of the objective functions. Finally, an "x" in Table 2 indicates that we could not infer the corresponding information from the results given in the reference.

The following observations can be made concerning the results of Table 2. First, as in the case of the non-minimax problems of Table 1, FSQP-NL performs better than FSQP-AL. This is especially true for the four nonlinearly constrained minimax problem (P43M to P117M) where the number of objective function evaluations and the number of constraint evaluations are both significantly lower with FSQP-NL. Second,

†FSQP Version 2.0 [19] gives the user the option to select either FSQP-AL or FSQP-NL.

while our algorithm was not specifically designed for minimax problems, it compares well with pure minimax algorithms proposed by others.

6. Concluding remarks. We have described and analyzed an SQP-type algorithm for inequality constrained optimization with the properties that *(i)* all iterates it generates satisfy the constraints and *(ii)* the Maratos effect is avoided while auxiliary constraint evaluations are performed only in the early iterations. We indicated how such algorithm can be adapted to handle constrained minimax problems. Numerical results measure up to the strong theoretical properties of the new algorithm.

It should be clear that in case equality constraints are also present, these can be handled via the introduction of a merit function as described in part I of this two-part paper. The overall algorithm will generate iterates that all satisfy the *inequality* constraints and, again, the Maratos effect will be avoided without the need for auxiliary function evaluations, except in the early iterations.

Appendix

Proof of Proposition 3.1.

Showing that $\{x_k\}$ is bounded can be done similarly to the proof of the Theorem in [5], using Assumption A2 and the monotonical decrease on f. For the remainder of the proof, the only difference with what is done in [5] consists in showing that if $t_k\langle\nabla f(x_k), d_k^0\rangle$ converges to zero on a subsequence, then, on that same subsequence, (i) $t_k d_k^0$ and (ii) $\|x_{k+1} - x_k\|$ also converge to zero. (i) obviously holds in view of (3.1), the feasibility of the iterates, (2.1) and the fact that $t_k \leq 1$. Now, since x_k is in a compact set, in view of the continuity of $d^1(\cdot)$, d_k^1 is bounded. Also, since v_k is bounded by $\|d_k^0\|$, in view of the constraints in $QP(x_k, H_k)$, of the properties of $d^1(\cdot)$ and of the definition of ρ_k^ℓ and ρ_k^g it follows that ρ_k^ℓ and ρ_k^g converge to zero whenever d_k^0 does. Therefore, $t_k d_k^\ell$ and $t_k d_k^g$ converge to zero and, since whenever it is defined $\tilde{d}_k$ satisfies $\|\tilde{d}_k\| \leq \|d_k^0\|$, we obtain $\|x_{k+1} - x_k\| \to 0$. □

Proof of Theorem 3.4.

Let x^* be an accumulation point and let $\{x_k\}_{k\in K}$ be a subsequence converging to x^*. In view of (2.1), we may also assume, without loss of generality, that the subsequence $\{H_k\}_{k\in K}$ converges to some symmetric positive definite matrix H^*. In that case, in view of the work in [4] (see also [6, Lemma 3.2]), the subsequence $\{d_k^0\}_{k\in K}$ converges to a vector d^{0*}. In order to conclude, we show that $d^{0*} = 0$, so that, the feasible point x^* (limit of feasible iterates) is a KKT point for (P). We now suppose that $d^{0*} \neq 0$ so that $\exists\ \underline{d} > 0$ s.t. $\|d_k^0\| \geq \underline{d}$, $k \in K$, and we show that, in that case, there exists $\underline{t} > 0$ such that, for all $k \in K$ at which a stepsize is computed in Step 1 *vii*, $t_k \geq \underline{t}$, contradicting Proposition 3.3. In view of (3.1), the feasibility of the successive iterates and of (2.1),

$$\langle\nabla f(x_k), d_k^0\rangle \leq -\sigma_1 \underline{d}^2$$

and, from the definition of d_k^g in Step 1 v,

$$\langle\nabla f(x_k), d_k^g\rangle \leq -\theta\sigma_1 \underline{d}^2. \tag{A.1}$$

Also, in view of the continuity of $d^1(\cdot)$ and (2.2) and the definition of d_k^g in Step 1 v, there exists $\underline{\rho} > 0$ such that, for $k \in K$ large enough it holds

$$g_j(x_k) + \langle\nabla g_j(x_k), d_k^g\rangle \leq -\underline{\rho}. \tag{A.2}$$

Showing that, on K, t_k is bounded away from zero by a positive number can be done similarly to the proof of Proposition 3.2 in [13] using (A.1), (A.2) and the fact that, in view of the continuity of $d^1(\cdot)$ and the definition of d_k^g and $\tilde{d}_k$ in Step 1 *v* and *vi*, $\|d_k^g\|$ and $\|\tilde{d}_k\|$ are bounded on K. □

Proof of Theorem 3.8

We first show that, for k large enough,

$$f(x_k + d_k^\ell) - \alpha\langle \nabla f(x_k), d_k^0\rangle \le f(x_{k-3}) \ . \tag{A.3}$$

While this proof has similarities with that of Theorem 3.8 in [14], it is given here for the reader's convenience. First, expanding $f(x_k + d_k^\ell)$ to first order about x^* and making use of the KKT conditions associated with x^* we may write

$$f(x_k + d_k^\ell) = f(x^*) - \sum_{j\in I(x^*)} \mu_j^* \langle \nabla g_j(x^*), x_k + d_k^\ell - x^*\rangle + O(\|x_k + d_k^\ell - x^*\|^2) \tag{A.4}$$

Observing that $g_j(x^*) = 0$ for $j \in I(x^*)$, expanding $g_j(x_k + d_k)$ about x^* for $j \in I(x^*)$ and substituting in (A.4) we obtain

$$f(x_k + d_k^\ell) = f(x^*) - \sum_{j\in I(x^*)} \mu_j^* g_j(x_k + d_k^\ell) + O(\|x_k + d_k^\ell - x^*\|^2) \ . \tag{A.5}$$

On the other hand, in view of Proposition 3.5 *i* the optimality conditions associated with $QP(x_k, H_k)$ yield, in view of (2.1),

$$\langle \nabla f(x_k), d_k^0\rangle = \sum_{j\in I(x^*)} \mu_{k,j} g_j(x_k) + O(\|d_k^0\|^2) \tag{A.6}$$

Also, in view of Proposition 3.5 *ii* the optimality conditions associated with $QP(x_k, H_k)$ and $QP(x_{k-1}, H_{k-1})$ respectively imply,

$$g_j(x_k + d_k^0) = O(\|d_k^0\|^2) \quad j \in I(x^*)$$

and

$$g_j(x_k) = O(\|d_{k-1}^0\|^2) \quad j \in I(x^*) \ .$$

In view of Proposition 3.5 *iii*, it thus follows from (A.5) and (A.6) that

$$\begin{aligned} &f(x_k + d_k^\ell) - \alpha\langle \nabla f(x_k), d_k^0\rangle \\ &= f(x^*) + O(\|x_k + d_k^\ell - x^*\|^2) + O(\|d_{k-1}^0\|^2) + O(\|d_k^0\|^2) \end{aligned}$$

since, in view of Proposition 3.5 *i*, the μ_k's are bounded. It then follows from Theorem 3.7 that

$$f(x_k + d_k^\ell) - \alpha\langle \nabla f(x_k), d_k^0\rangle = f(x^*) + o(\|x_{k-3} - x^*\|^2) \ .$$

Equation (A.3) then follows from the easily checked fact [1, Lemma 1] that there exists a positive scalar C such that for x feasible close enough to x^*,

$$f(x) \ge f(x^*) + C\|x - x^*\|^2 \ .$$

The theorem follows from (A.3) and the fact that, in view of Proposition 3.6, for k large enough,

$$g_j(x_k + d_k^\ell) \le 0, \quad j = 1, \ldots, m.$$

□

REFERENCES

[1] R. M. CHAMBERLAIN, M. J. D. POWELL, C. LEMARECHAL, and H. C. PEDERSEN, *The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization*, Math. Programming Stud., 16 (1982), pp. 1–17.

[2] C. CHARALAMBOUS AND A. R. CONN, *An Efficient Method to Solve the Minimax Problem Directly*, SIAM J. Numer. Anal., 15 (1978), pp. 162–187.

[3] A. R. CONN AND Y. LI, *An Efficient Algorithm for Nonlinear Minimax Problems*, University of Waterloo, Research Report CS-88-41, Waterloo, Ontario, N2L 3G1 Canada, November, 1989 .

[4] J. M. DANIEL, *Stability of the Solution of Definite Quadratic Programs*, Math. Programming, 5 (1970), pp. 41–53.

[5] L. GRIPPO, F. LAMPARIELLO, and S. LUCIDI, *A Nonmonotone Line Search Technique for Newton's Method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[6] S. P. HAN, *A Globally Convergent Method for Nonlinear Programming*, J. Optim. Theory Appl., 22 (1977), pp. 297–309.

[7] ———, *Superlinear Convergence of a Minimax Method*, Department of Computer Science, Cornell University, TR78-336, 1978.

[8] ———, *Variable Metric Methods for Minimizing a Class of Nondifferentiable Functions*, Math. Prog., 20 (1981), pp. 1–13.

[9] W. HOCK AND K. SCHITTKOWSKI, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems (187), Springer Verlag, 1981.

[10] K. MADSEN AND H. SCHJÆR-JACOBSEN, *Linearly Constrained Minimax Optimization*, Mathematical Programming, 14 (1978), pp. 208–223.

[11] N. MARATOS, *Exact Penalty Function Algorithms for Finite Dimensional and Optimization Problems*, Ph.D. Thesis, Imperial College of Science and Technology, London, U.K., 1978.

[12] D. Q. MAYNE AND E. POLAK, *A Superlinearly Convergent Algorithm for Constrained Optimization Problems*, Math. Programming Stud., 16 (1982), pp. 45–61.

[13] E. R. PANIER AND A. L. TITS, *A Superlinearly Convergent Feasible Method for the Solution of Inequality Constrained Optimization Problems*, SIAM J. Control Optim., 25 (1987), pp. 934–950.

[14] ———, *Avoiding the Maratos Effect by Means of a Nonmonotone Line Search. I. General Constrained Problems*, Systems Research Center, University of Maryland, Technical Report SRC-TR-89-41, College Park, MD 20742, 1989.

[15] ———, *On Feasibility, Descent and Superlinear Convergence in Inequality Constrained Optimization*, Systems Research Center, University of Maryland, Technical Report SRC-TR-89-27, College Park, MD 20742, 1989.

[16] ———, *A Superlinearly Convergent Method of Feasible Directions for Optimization Problems Arising in the Design of Engineering Systems*, in Proc. of the Seventh International Conference on Analysis and Optimization of Systems — Antibes, June 25-27, 1986, A. Bensoussan and J. L. Lions, eds., Lecture Notes in Control and Information Sciences, 83, Springer Verlag, Berlin-Heidelberg-New York-Tokyo, June 1986, pp. 65–73.

[17] M. J. D. POWELL, *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*, in Numerical Analysis, Dundee, 1977, Lecture Notes in Mathematics 630, G. A. Watson, ed., Springer-Verlag, 1978, pp. 144–157.

[18] G. A. WATSON, *The Minimax Solution of an Overdetermined System of Nonlinear Equations*, J. Inst. Math. Appl., 23 (1979), pp. 167–180.

[19] J. ZHOU AND A. L. TITS, *User's Guide for FSQP Version 2.0 – A Fortran Code for Solving Optimization Problems, Possibly Minimax, with General Inequality Constraints and Linear Equality Constraints, Generating Feasible Iterates*, Systems Research Center, University of Maryland, SRC TR-90-60, College Park, MD 20742, 1990.

PROB	CODE	NNL	NF	NG	ITER	OBJECTIVE	KKT	EPS
P12	FSQP-AL	1	7	15	7	−.300000000E+02	.72E-06	.10E-05
	FSQP-NL		7	13	7	−.300000000E+02	.79E-06	.10E-05
P29	FSQP-AL	1	12	23	11	−.226274170E+02	.13E-07	.10E-05
	FSQP-NL		13	17	13	−.226274170E+02	.19E-06	.10E-05
P30	FSQP-AL	1	16	31	16	.100000000E+01	.54E-08	.10E-07
	FSQP-NL		15	15	15	.100000000E+01	.97E-08	.10E-07
P31	FSQP-AL	1	9	21	8	.600000000E+01	.23E-05	.10E-04
	FSQP-NL		10	19	10	.600000000E+01	.46E-06	.10E-04
P32	FSQP-AL	1	3	6	3	.100000000E+01	.31E-15	.10E-07
	FSQP-NL		3	4	3	.100000000E+01	.31E-15	.10E-07
P33	FSQP-AL	2	4	14	4	−.400000000E+01	.13E-11	.10E-07
	FSQP-NL		5	10	5	−.400000000E+01	.47E-11	.10E-07
P34	FSQP-AL	2	7	28	7	−.834032443E+00	.19E-08	.10E-07
	FSQP-NL		9	24	9	−.834032445E+00	.38E-09	.10E-07
P43	FSQP-AL	3	11	62	9	−.440000000E+02	.12E-05	.10E-04
	FSQP-NL		13	55	13	−.440000000E+02	.86E-06	.10E-04
P51	FSQP-AL	0	8	0	6	.505655658E−15	.46E-06	.10E-05
	FSQP-NL		9		8	.505655658E−15	.34E-08	.10E-05
P57	FSQP-AL	1	7	9	3	.306463061E−01	.29E-05	.10E-04
	FSQP-NL		7	8	3	.306463061E−01	.28E-05	.10E-04
P66	FSQP-AL	2	8	30	8	.518163274E+00	.50E-09	.10E-07
	FSQP-NL		9	24	9	.518163274E+00	.63E-11	.10E-07
P76	FSQP-AL	0	6	0	6	−.468181818E+01	.34E-04	.10E-03
	FSQP-NL		6		6	−.468181818E+01	.34E-04	.10E-03
P84	FSQP-AL	6	4	42	4	−.528033513E+07	.68E-12	.10E-08
	FSQP-NL		4	30	4	−.528033513E+07	.66E-09	.10E-08
P86	FSQP-AL	0	14	0	9	−.323486790E+02	.17E-13	.10E-07
	FSQP-NL		8		7	−.323486790E+02	.17E-13	.10E-07
P93	FSQP-AL	2	15	61	12	.135075968E+03	.37E-03	.10E-02
	FSQP-NL		15	38	15	.135075964E+03	.41E-04	.10E-02
P100	FSQP-AL	4	23	168	16	.680630057E+03	.62E-06	.10E-03
	FSQP-NL		20	128	17	.680630057E+03	.26E-04	.10E-03
P110	FSQP-AL	0	10	0	9	−.457784697E+02	.86E-10	.10E-07
	FSQP-NL		10		9	−.457784697E+02	.86E-10	.10E-07
P113	FSQP-AL	5	12	122	12	.243063768E+02	.81E-03	.10E-02
	FSQP-NL		12	106	12	.243064357E+02	.85E-03	.10E-02
P117	FSQP-AL	5	20	219	19	.323486790E+02	.58E-04	.10E-03
	FSQP-NL		18	94	17	.323486790E+02	.34E-04	.10E-03
P118	FSQP-AL	0	19	0	19	.664820450E+03	.13E-14	.10E-07
	FSQP-NL		19		19	.664820450E+03	.13E-14	.10E-07

NNL: number of nonlinear constraints.
NF: number of objective function evaluations.
NG: number of (scalar) constraint evaluations.
ITER: number of iterations.
OBJECTIVE: objective function value at the final iterate.
KKT: norm of KKT vector (the gradient of the Lagrangian) at the final iterate.
EPS: maximum allowed for KKT (stopping criterion).

Table 1

PROB	CODE	NOBJ	NNL	NMF	NG	ITER	OBJMAX	KKT	EPS
BARD	NM	15	0	10	0	x	x	x	.50E-05
	FSQP-AL			15		8	.508163265E−01	.63E-10	.50E-05
	FSQP-NL			7		7	.508168686E−01	.42E-05	.50E-05
CB2	NM	3	0	11	0	x	x	x	.50E-05
	FSQP-AL			11		6	.195222453E+01	.10E-06	.50E-05
	FSQP-NL			6		6	.195222453E+01	.82E-06	.50E-05
CB3	NM	3	0	6	0	x	x	x	.50E-05
	FSQP-AL			5		3	.200000000E+01	.75E-06	.50E-05
	FSQP-NL			5		5	.200000000E+01	.94E-09	.50E-05
COLV	NM	6	0	49	0	x	x	x	.50E-05
	FSQP-AL			31		15	.274053332E+02	.11E-05	.50E-05
	FSQP-NL			14		14	.274053332E+02	.14E-05	.50E-05
DAVD2	NM	20	0	20	0	x	x	x	.50E-05
	FSQP-AL			20		10	.115706440E+03	.59E-06	.50E-05
	FSQP-NL			11		10	.115706440E+03	.93E-06	.50E-05
F&R	NM	2	0	11	0	x	x	x	.50E-05
	FSQP-AL			17		9	.494895210E+01	.24E-06	.50E-05
	FSQP-NL			10		10	.494895210E+01	.21E-06	.50E-05
HETTICH	NM	5	0	11	0	x	x	x	.50E-05
	FSQP-AL			19		10	.245935695E−02	.28E-05	.50E-05
	FSQP-NL			11		10	.245939485E−02	.19E-05	.50E-05
R-S	NM	4	0	12	0	x	x	x	.50E-05
	FSQP-AL			22		9	−.440000000E+02	.13E-05	.50E-05
	FSQP-NL			16		10	−.440000000E+02	.99E-07	.50E-05
WATS-6	NM	31	0	24	0	x	x	x	.50E-05
	FSQP-AL			23		12	.127170954E−01	.14E-05	.50E-05
	FSQP-NL			14		13	.127170913E−01	.31E-08	.50E-05
WATS-20	NM	31	0	22	0	x	x	x	.50E-05
	FSQP-AL			106		42	.138908355E−07	.35E-06	.50E-05
	FSQP-NL			45		43	.141191856E−07	.17E-06	.50E-05
WONG	NM	5	0		0		x	x	.50E-05
	FSQP-AL			67		20	.680630057E+03	.12E-05	.50E-05
	FSQP-NL			49		26	.680630057E+03	.42E-05	.50E-05
MAD1	MS	3	0	x	0	8	x	x	.10E-11
	FSQP-AL			9		5	−.389659516E+00	.35E-16	.10E-11
	FSQP-NL			6		6	−.389659516E+00	.89E-10	.10E-11
MAD2	MS	3	0	x	0	x	x	x	.10E-11
	FSQP-AL			21		11	−.330357143E+00	.13E-10	.10E-11
	FSQP-NL			19		18	−.330357143E+00	.81E-10	.10E-11
MAD4	MS	3	0	x	0	8	x	x	.10E-11
	FSQP-AL			11		6	−.448910786E+00	.90E-16	.10E-11
	FSQP-NL			8		8	−.448910786E+00	.90E-16	.10E-11
MAD5	MS	3	0	x	0	8	x	x	.10E-11
	FSQP-AL			13		7	−.100000000E+01	.16E-16	.10E-11
	FSQP-NL			8		8	−.100000000E+01	.35E-13	.10E-11
MAD6	MS	163	0	8	0	x	.113105 E+00	x	.10E-11
	FSQP-AL			11		6	.113104635E+00	.20E-10	.10E-11
	FSQP-NL			8		8	.113104727E+00	.72E-15	.10E-11
MAD8-10	MS	18	0	14	0	x	x	x	.10E-11
	FSQP-AL			19		10	.381173963E+00	.99E-12	.10E-11
	FSQP-NL			14		14	.381173963E+00	.22E-15	.10E-11
MAT8-30	MS	58	0	15	0	x	x	x	.10E-11
	FSQP-AL			30		15	.547620496E+00	.21E-15	.10E-11
	FSQP-NL			20		18	.547620496E+00	.21E-10	.10E-11
MAT8-50	MS	98	0	15	0	x	x	x	.10E-11
	FSQP-AL			39		20	.579276202E+00	.20E-15	.10E-11
	FSQP-NL			21		21	.579276202E+00	.22E-13	.10E-11
P43M	FSQP-AL	3	1	27	36	14	−.440000000E+02	.30E-06	.50E-05
	FSQP-NL			20	25	16	−.440000000E+02	.39E-05	.50E-05
P84M	FSQP-AL	3	4	7	28	4	−.528033513E+07	.0	.50E-05
	FSQP-NL			3	12	3	−.528033513E+07	.37E-03	.50E-05
P113M	FSQP-AL	4	5	25	142	13	.243062091E+02	.31E-05	.50E-05
	FSQP-NL			21	115	15	.243062091E+02	.31E-05	.50E-05
P117M	FSQP-AL	3	3	48	124	21	.323486790E+02	.46E-05	.50E-05
	FSQP-NL			19	54	17	.323486790E+02	.26E-04	.50E-05

Table 2