

TECHNICAL RESEARCH REPORT

Nonlinear Equality Constraints in Feasible Sequential Quadratic Programming

by C.T. Lawrence and A.L. Tits

T.R. 95-2



*Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry*

Nonlinear Equality Constraints in Feasible Sequential Quadratic Programming*

Craig T. Lawrence and André L. Tits

*Institute for Systems Research and
Electrical Engineering Department
University of Maryland, College Park
College Park, MD 20742*

Received January, 1995

A simple scheme is proposed for handling nonlinear equality constraints in the context of a previously introduced sequential quadratic programming (SQP) algorithm for inequality constrained problems, generating iterates satisfying all constraints. The key is an idea due to Mayne and Polak (*Math. Progr.*, vol. 11, pp. 67–80, 1976) by which nonlinear equality constraints are treated as “ \leq ”-type constraints to be satisfied by all iterates, thus precluding any positive value, and an exact penalty term is added to the objective function which penalizes negative values. Mayne and Polak obtain a suitable value of the penalty parameter by iterative adjustments based on a test involving estimates of the KKT multipliers. We argue that the SQP framework allows for a more effective estimation of these multipliers, and we provide convergence analysis of the resulting algorithm. Numerical results, obtained with the FSQP/CFSQP code, are reported.

KEY WORDS: Constrained optimization, nonlinear equality constraints, sequential quadratic programming, feasibility

1 INTRODUCTION

Notation. Let \mathbb{R} denote the set of real numbers, \mathbb{R}^n the set of real n -vectors, and \mathbb{N} the set of natural numbers. Given $x \in \mathbb{R}^n$, x^j denotes the j th component of the vector x . Given two vectors $x, y \in \mathbb{R}^n$, $\langle x, y \rangle$ denotes the standard inner product on \mathbb{R}^n and $\|x\|$ the standard Euclidian norm. To indicate that a symmetric matrix $H \in \mathbb{R}^{n \times n}$ is positive definite, we write $H > 0$. The notation $\{x_k\}_{k \in \mathcal{K}}$ is used to denote a sequence of vectors $x_k \in \mathbb{R}^n$ with indices in the index set \mathcal{K} . Finally, we write $x_k \xrightarrow{k \in \mathcal{K}} x^*$ to indicate that the sequence converges to x^* on the infinite index set \mathcal{K} .

* This research was supported in part by NSF grant No. DMI-93-13286 and by the NSF Engineering Research Centers Program NSFD-CDR-88-03012.

Consider the problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_0(x) \\ \text{s.t.} \quad & f_j(x) = 0, \quad j = 1, \dots, m_e, \\ & g_j(x) \leq 0, \quad j = 1, \dots, m_i, \end{aligned} \tag{P}$$

where $f_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 0, 1, \dots, m_e$, and $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, m_i$ are continuously differentiable. In the case when $m_e = 0$, a recently proposed “feasible” sequential quadratic programming (FSQP) algorithm [7] efficiently solves such problems while forcing all iterates to remain feasible (i.e., to satisfy all constraints). Advantages of feasible iterates are discussed in [1, 7]. While equality constraints can easily be handled by means of a standard quadratic penalty function, the feasible iterate framework makes it possible to use a more satisfactory scheme proposed by Mayne and Polak in the context of first order methods of feasible directions [6] (and later used by Herskovits in [2]). Their scheme considers the related family of inequality constrained problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_0(x) - c \sum_{j=1}^{m_e} f_j(x) \\ \text{s.t.} \quad & f_j(x) \leq 0, \quad j = 1, \dots, m_e, \\ & g_j(x) \leq 0, \quad j = 1, \dots, m_i, \end{aligned} \tag{\tilde{P}_c}$$

where $c > 0$. It is clear that large positive values of c penalize iterates satisfying $f_j(x) < 0$ for some $j \in \{1, \dots, m_e\}$, while feasibility for the modified problem enforces $f_j(x) \leq 0$ for all $j \in \{1, \dots, m_e\}$. Intuitively then, the sequence of iterates generated by a feasible direction algorithm for the modified problem should tend towards feasibility for the original problem. The key advantage of this scheme is that the modified objective function is an *exact* differentiable penalty function (i.e. the solution of (\tilde{P}_c) corresponds to that of (P) for large enough, but finite, c) when the problem is solved via an algorithm generating feasible iterates.

The essence of the approach used by Mayne and Polak is to iteratively solve (\tilde{P}_c) generating a sequence of points x_k feasible for the modified problem, while simultaneously increasing the parameter c until it is large enough to guarantee that any accumulation point of the sequence $\{x_k\}_{k \in \mathbb{N}}$ lies in the feasible set for (P) . In [6], Mayne and Polak show that convergence for the original problem is guaranteed when the penalty parameter is updated in such a way that it is eventually larger than the largest magnitude of an equality constraint Karush-Kuhn-Tucker (KKT) multiplier at the solution. This suggests that a reasonable update scheme may be to increase the penalty parameter whenever an estimate of a multiplier at the current iterate exceeds the current penalty parameter in magnitude.

In order to estimate the multipliers at a point x_k , Mayne and Polak suggest solving a least squares problem. That is, the multiplier estimates are computed as the coefficients of the projection of $\nabla f_0(x_k)$ into the space spanned by the equality and active inequality constraint gradients. However, nonlinear constraints active at

the solution might not be active at any iterate. For this reason, Mayne and Polak include in their least squares problem the gradients of all inequality constraints whose absolute value is less than a certain parameter $\epsilon' > 0$. The appropriate choice of this parameter is not at all clear. If ϵ' is chosen too large (resp. too small) the set of active constraints may be overestimated (resp. underestimated), possibly resulting in an inappropriate value of the penalty parameter c_k . Of course, if ϵ' is small enough, the correct active set will eventually be identified, but progress may be slow in early iterations. Fortunately, as discussed below, more satisfactory alternatives are available in the context of second-order feasible direction methods, such as the algorithm proposed in [7].

In SQP-type methods, a candidate search direction d^0 is obtained as the solution of a quadratic program approximating (to second order) the original nonlinear program around the current iterate. Given $c > 0$, define the modified objective function

$$\phi_c(x) \triangleq f_0(x) - c \sum_{j=1}^{m_e} f_j(x).$$

Let $x \in \mathbb{R}^n$ be the current iterate and let H be a symmetric positive definite approximation to the Hessian of the Lagrangian for (\tilde{P}_c) at x . Then the SQP direction $d^0 = d^0(x, c, H)$ for the problem (\tilde{P}_c) is defined as the solution of the quadratic program

$$\begin{aligned} \min_{d^0 \in \mathbb{R}^n} \quad & \frac{1}{2} \langle d^0, H d^0 \rangle + \langle \nabla \phi_c(x), d^0 \rangle \\ \text{s.t.} \quad & f_j(x) + \langle \nabla f_j(x), d^0 \rangle \leq 0, \quad j = 1, \dots, m_e, \\ & g_j(x) + \langle \nabla g_j(x), d^0 \rangle \leq 0, \quad j = 1, \dots, m_i. \end{aligned} \tag{1}$$

In relation with the Mayne and Polak scheme, a key by-product of the solution of the quadratic program is a vector of KKT multiplier estimates.

However, the multipliers obtained from (1) associated with the linearizations of the f_j 's in (\tilde{P}_c) cannot be directly used to determine the next value of the penalty parameter c . Indeed, they are zero whenever the corresponding f_j is significantly negative, which is clearly a situation that would indicate increasing c may be in order. Other alternatives are available, though. Suppose we are to generate a sequence $\{x_k\}_{k \in \mathbb{N}}$. For each k , denote by $\lambda_k^j, j = 1, \dots, m_i$, the KKT multipliers from (1), with $x = x_k$, associated with the linearization of g_j . We could use this information in one of two ways. First, we could simply deem a constraint g_j "active" for (\tilde{P}_c) when $\lambda_k^j > 0$ and solve the least squares problem as in Mayne and Polak's scheme. This eliminates the need for the extra parameter ϵ' . Alternatively, we could make further use of the λ_k^j 's and estimate the equality constraint multipliers at the *next* iterate x_{k+1} as the coefficients of the projection of $\nabla f_0(x_{k+1}) + \sum_{j=1}^{m_i} \lambda_k^j \nabla g_j(x_{k+1})$ onto the space spanned by the gradients of the equality constraints at x_{k+1} . That is, use the multipliers from the computation of the SQP direction d^0 at x_k as our inequality constraint multiplier estimates at x_{k+1} , and solve for the equality constraint multiplier estimates through the least squares problem. This also eliminates the need for the parameter ϵ' , and further, the size of the least squares problem is

reduced. In this paper we investigate incorporating into the FSQP algorithm the Mayne and Polak scheme modified along the lines of this second alternative.

The balance of this paper is organized as follows. In Section 2 we present the algorithm (a few of the details are deferred to Section 5 in order to avoid any loss of continuity). In Section 3 we discuss convergence of the algorithm for the special case when $m_i = 0$. Section 4 is devoted to establishing convergence for the general case. In Section 5 we discuss an implementation and some numerical results. Finally, we offer some concluding remarks in Section 6.

2 ALGORITHM

Let

$$\Omega \triangleq \{ x \in \mathbb{R}^n \mid f_j(x) = 0, j = 1, \dots, m_e, g_j(x) \leq 0, j = 1, \dots, m_i \}$$

be the feasible set for the problem (P), and

$$\tilde{\Omega} \triangleq \{ x \in \mathbb{R}^n \mid f_j(x) \leq 0, j = 1, \dots, m_e, g_j(x) \leq 0, j = 1, \dots, m_i \}$$

be the feasible set for the problem (\tilde{P}_c). Note that $\Omega \subset \tilde{\Omega}$. We make the following assumptions:

Assumption 1. The feasible set $\tilde{\Omega}$ for the modified problem (\tilde{P}_c) has a nonempty interior.

Assumption 2. For all $x \in \mathbb{R}^n$, the vectors $\nabla f_j(x)$, $j = 1, \dots, m_e$, and $\nabla g_j(x)$, $j \in \{j \mid g_j(x) = 0\}$ are linearly independent.

A point $x^* \in \Omega$ is said to be a *Karush-Kuhn-Tucker* (KKT) point for problem (P) if there exist multipliers ψ_j^* , $j = 1, \dots, m_e$, and $\lambda_j^* \geq 0$, $j = 1, \dots, m_i$, such that

$$\nabla f_0(x^*) + \sum_{j=1}^{m_e} \psi_j^* \nabla f_j(x^*) + \sum_{j=1}^{m_i} \lambda_j^* \nabla g_j(x^*) = 0,$$

$$f_j(x^*) = 0, \quad j = 1, \dots, m_e,$$

$$g_j(x^*) \leq 0, \quad j = 1, \dots, m_i,$$

and the complementary slackness conditions

$$\lambda_j^* g_j(x^*) = 0, \quad j = 1, \dots, m_i,$$

are satisfied. Similarly, a point $x^* \in \tilde{\Omega}$ is a KKT point for (\tilde{P}_c) if there exist

nonnegative multipliers $\tilde{\psi}_j^*$, $j = 1, \dots, m_e$, and $\tilde{\lambda}_j^*$, $j = 1, \dots, m_i$, satisfying

$$\begin{aligned} \nabla \phi_c(x^*) + \sum_{j=1}^{m_e} \tilde{\psi}_j^* \nabla f_j(x^*) + \sum_{j=1}^{m_i} \tilde{\lambda}_j^* \nabla g_j(x^*) &= 0, \\ f_j(x^*) \leq 0, \quad \tilde{\psi}_j^* f_j(x^*) &= 0, \quad j = 1, \dots, m_e, \\ g_j(x^*) \leq 0, \quad \tilde{\lambda}_j^* g_j(x^*) &= 0, \quad j = 1, \dots, m_i. \end{aligned}$$

The following proposition is proved in [6].

PROPOSITION 2.1. *If $x^* \in \Omega$ is a KKT point for (\tilde{P}_c) , then x^* is a KKT point for (P) .*

Algorithm 1 below is an extension of the algorithm given in [7]. As suggested in Section 1, consider solving the problem (\tilde{P}_c) for a fixed $c > 0$. Let $x \in \mathbb{R}^n$ be the current iterate and let H be a symmetric positive definite approximation to the Hessian of the Lagrangian for (\tilde{P}_c) at x . In order to construct a new iterate in such a way that local superlinear convergence is guaranteed, and feasibility for the modified problem is maintained, the algorithm proposed in [7] performs a search along an arc defined by three direction vectors. The first direction is the standard SQP direction, which we call $d^0 = d^0(x, c, H)$, and is defined as the solution of the QP (1). The solution d^0 may not yield a feasible search direction for (\tilde{P}_c) . In order to generate a feasible direction, while staying as close to d^0 as possible, the SQP direction is “tilted” via a strictly feasible descent direction $d^1 = d^1(x, c, d^0)$. More precisely, $d^1(\cdot, \cdot, \cdot)$ is a continuous map constructed so that $d^1(x, c, 0) = 0$ if x is a KKT point for (\tilde{P}_c) and is a strictly feasible descent direction if x is not a KKT point (see [7]). The corrected direction d is formed as a convex combination of d^0 and d^1 , i.e. $d = (1 - \rho)d^0 + \rho d^1$ where $\rho = \rho(d^0) \in [0, 1]$. Following [7], $\rho(\cdot) : \mathbb{R}^n \rightarrow [0, 1]$ is defined in such a way that

$$\rho(d^0) = O(\|d^0\|^2). \quad (2)$$

Finally, in order to avoid the Maratos effect (and guarantee local superlinear convergence), a direction $\tilde{d} = \tilde{d}(x, c, d, H)$ is computed. The correction \tilde{d} is chosen so that close to a solution $x + d + \tilde{d}$ is feasible, $\phi_c(x + d + \tilde{d}) < \phi_c(x)$, and $d + \tilde{d}$ converges to d (again, see [7]). An Armijo-type search is then performed along the arc $x + td + t^2\tilde{d}$ for $t \in (0, 1]$.

As suggested above, the multiplier estimates to be used for updating the penalty parameter c are obtained as follows. The objective gradient plus the sum of the inequality constraint gradients times their multiplier estimates (as obtained from the solution of (1)) is projected into the space spanned by equality constraint gradients. That is, we solve the following least squares problem for $\bar{\mu} = \bar{\mu}(x, \lambda) \in \mathbb{R}^{m_e}$:

$$\min_{\bar{\mu} \in \mathbb{R}^{m_e}} \left\| \nabla f_0(x) + \sum_{j=1}^{m_i} \lambda^j \nabla g_j(x) + \sum_{j=1}^{m_e} \bar{\mu}^j \nabla f_j(x) \right\|^2, \quad (3)$$

where λ^j , $j = 1, \dots, m_i$ are the multipliers corresponding to the linearizations of g_j in the QP (1). In view of Assumption 2, the associated Gram matrix for the least squares problem is positive definite, hence the following proposition must hold.

PROPOSITION 2.2. *Under the current assumptions, the solution $\bar{\mu}(x, \lambda)$ of (3) is unique and continuous as a function of x and λ .*

The penalty parameter c is compared against the most negative of the equality constraint multiplier estimates and, if necessary, updated so that it is larger in magnitude than this multiplier. We are now ready to state the algorithm.

Algorithm 1.

Parameters. $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$, $\delta > 1$, $\gamma > 0$, $M > 0$.

Data. $x_0 \in \tilde{\Omega}$, $H_0 \in \mathbb{R}^{n \times n}$ where $H_0 = H_0^T > 0$, $c_0 > 0$.

Step 0: Initialization. Set $k = 0$.

Step 1: Computation of a search arc.

- i. Compute $d_k^0 = d^0(x_k, c_k, H_k)$ and the multipliers λ_k^j , $j = 1, \dots, m_i$.
If $d_k^0 = 0$ and $\sum_{j=1}^{m_e} |f_j(x_k)| = 0$ stop.
- ii. Compute $d_k^1 = d^1(x_k, c_k, d_k^0)$.
- iii. Compute $\rho_k = \rho(d_k^0)$ and set $d_k = (1 - \rho_k)d_k^0 + \rho_k d_k^1$.
- iv. Compute $\tilde{d}_k = \tilde{d}(x_k, c_k, d_k, H_k)$. If $\|\tilde{d}_k\| > \|d_k\|$, set $\tilde{d}_k = 0$.

Step 2. Arc search. Compute t_k , the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$\begin{aligned} \phi_{c_k}(x_k + td_k + t^2\tilde{d}_k) &\leq \phi_{c_k}(x_k) + \alpha t \langle \nabla \phi_{c_k}(x_k), d_k \rangle, \\ f_j(x_k + td_k + t^2\tilde{d}_k) &\leq 0, \quad j = 1, \dots, m_e, \\ g_j(x_k + td_k + t^2\tilde{d}_k) &\leq 0, \quad j = 1, \dots, m_i. \end{aligned}$$

Step 3. Updates.

- i. Set $x_{k+1} = x_k + t_k d_k + t_k^2 \tilde{d}_k$.
- ii. Compute $\bar{\mu}_k = \bar{\mu}(x_{k+1}, \lambda_k)$.
- iii. Update the penalty parameter,

$$c_{k+1} = \begin{cases} c_k & \text{if } c_k + \min_j \{\bar{\mu}_k^j\} \geq \gamma \text{ or } c_k \|H_k d_k^0\| \geq M, \\ \max\{\gamma - \min_j \{\bar{\mu}_k^j\}, \delta c_k\} & \text{else.} \end{cases}$$

- iv. Compute a new symmetric positive definite approximation H_{k+1} to the Hessian of the Lagrangian of $(\tilde{P}_{c_{k+1}})$.
- v. Set $k = k + 1$.

Go back to *Step 1*. \square

It will be shown below (Lemma 4.2) that, if $d_k^0 = 0$ and if c_k happens to be larger than the largest absolute value of the components of $\bar{\mu}(x_k, \lambda_k)$, then x_k must be feasible for (P) . However, in Algorithm 1, the value of c_k is based on the value of $\bar{\mu}(x_k, \lambda_{k-1})$, i.e., it depends on the QP multipliers at the previous iteration (for good reason: the value of c_k must be known in order to solve the QP at the current iteration). This is the reason why feasibility must be checked explicitly in the stopping criterion in *Step 1(i)*. If $d_k^0 = 0$ and the equality constraints are not satisfied, no step is taken. The penalty parameter is then recomputed using the *current* multipliers and the algorithm again attempts to construct a search direction. Using the updated penalty parameter, the SQP direction will be nonzero and the algorithm will move away from the infeasible point.

Finally, a word of explanation is in order concerning the condition under which c_k is updated. Namely, in order to guarantee that c_k remains bounded, it is necessary to add a condition to the test in *Step 3(iii)* for increasing the penalty parameter that was not needed in [6] (or in the special case we consider in Section 3). Without such a test, it may happen that the updated c_k leads to large multipliers λ_k in the QP for d_k^0 , in turn forcing another increase of c_k . This could result in a “run-away” phenomenon with $\{c_k\}_{k \in \mathbb{N}}$ diverging to infinity.

We make one further standard assumption, this one concerning the approximations to the Hessian of the Lagrangian as computed in *Step 3(iv)* of Algorithm 1.

Assumption 3. If the sequence $\{c_k\}_{k \in \mathbb{N}}$ generated by Algorithm 1 is bounded, then there exists constants $\sigma_2 \geq \sigma_1 > 0$ such that

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

for all $k \in \mathbb{N}$.

If c_k is not updated, i.e. if it is kept fixed at a constant value \bar{c} , then Algorithm 1 reduces to the algorithm in [7] applied to the problem $(\bar{P}_{\bar{c}})$. Thus, the following result follows directly from Propositions 3.1, 3.2, and 3.3 in [7], proved under the current assumptions.

PROPOSITION 2.3. *Algorithm 1 is well-defined (i.e. Step 2 is well-defined). Moreover, (i) given $c > 0$, $x \in \tilde{\Omega}$, and $H = H^T > 0$, $d^0(x, c, H) = 0$ if, and only if, x is a KKT point for (\bar{P}_c) , and (ii) if the algorithm never stops in Step 1(i), and if $\{c_k\}_{k \in \mathbb{N}}$ is eventually constant, say $c_k = \bar{c}$ for all k large enough, then every accumulation point of $\{x_k\}_{k \in \mathbb{N}}$ is a KKT point for $(\bar{P}_{\bar{c}})$.*

3 A SPECIAL CASE

In this section we investigate the convergence of Algorithm 1 for the solution of a problem with only equality constraints, i.e. $m_i = 0$ in (P) . As just pointed out, this assumption removes the need to check that $c_k \|H_k d_k^0\| < M$ before updating the penalty parameter. It also greatly simplifies the convergence analysis. The results

in this section essentially follow from the work of Mayne and Polak [6], but we give an alternative, more direct, development. We show that any accumulation point of the sequence of iterates generated via Algorithm 1 is a KKT point for problem (P) with $m_i = 0$.

Consider the case in which Algorithm 1 generates a finite sequence $\{x_1, \dots, x_N\}$.

PROPOSITION 3.1. *Suppose Algorithm 1 generates a finite sequence with final iterate x_N . Then x_N is a KKT point for (P).*

Proof. From the stopping criterion in *Step 1(i)* we see that $d^0(x_N, c_N, H_N) = 0$ and $f_j(x_N) = 0$, $j = 1, \dots, m_e$. Thus, Proposition 2.3 tells us that x_N is a KKT point for (\tilde{P}_{c_N}) . Finally, we conclude from Proposition 2.1 that x_N is a KKT point for (P). \square

Suppose now that Algorithm 1 generates an infinite sequence $\{x_k\}_{k \in \mathbb{N}}$. By construction, for all k , $f_j(x_k) \leq 0$, $j = 1, \dots, m_e$. In order for all accumulation points x^* to satisfy the constraints for our problem, c_k must be chosen in such a way that $\|d^0(x_k, c_k, H_k)\|$ is bounded away from zero unless x_k approaches a feasible point for (P). The following lemma, which is similar to Theorem 1 in Mayne and Polak [6], shows that this is indeed the case (recall that $d^0(\cdot, \cdot, \cdot)$ is continuous).

LEMMA 3.2. *Let $x \in \tilde{\Omega}$, $H = H^T > 0$. Suppose $\bar{\mu} \in \mathbb{R}^{m_e}$ is the solution of the least squares problem (3) and $c > 0$ satisfies $c > -\bar{\mu}^j$, $j = 1, \dots, m_e$. If $d^0(x, c, H) = 0$, then $f^j(x) = 0$, $j = 1, \dots, m_e$, i.e. $x \in \Omega$.*

This lemma is not proved here, as it is a simplified version of a lemma that is proved in the next section. We now show that, under an additional assumption, the penalty parameter c_k is increased only finitely many times. To avoid any ambiguity, let $\bar{\mu}(x)$ denote the solution of (3) with $m_i = 0$.

LEMMA 3.3. *If the infinite sequence $\{x_k\}_{k \in \mathbb{N}}$ generated by Algorithm 1 is bounded, then c_k is increased only finitely many times.*

Proof. By contradiction. Suppose c_k is increased infinitely many times. Then, since $\delta > 1$, we see from *Step 3(iii)* of Algorithm 1 that $c_k \rightarrow \infty$ and there exists an infinite index set \mathcal{K} such that

$$c_k + \min_j \{\bar{\mu}^j(x_k)\} < \gamma, \quad \forall k \in \mathcal{K}.$$

Thus, $\{\bar{\mu}(x_k)\}_{k \in \mathbb{N}}$ is an unbounded sequence. This is a contradiction, as $\bar{\mu}(x)$ is a continuous function of x (by Proposition 2.2 with $m_i = 0$), and the sequence $\{x_k\}_{k \in \mathbb{N}}$ is bounded. \square

When attempting to prove the preceding lemma for the general case, i.e. with inequality constraints, the additional test is required in the penalty parameter update. The details are presented in the next section. The main result for the special case is as follows.

THEOREM 3.4. *If Algorithm 1 generates a bounded infinite sequence $\{x_k\}_{k \in \mathbb{N}}$, then any accumulation point x^* is a KKT point for (P) .*

Proof. In view of Lemma 3.3, as $\{x_k\}_{k \in \mathbb{N}}$ is bounded, we know that c_k is increased only finitely many times. Suppose it is eventually fixed at \bar{c} , that is $c_k = \bar{c}$ for all $k \geq N$ where N is finite. Proposition 2.3 tells us that an accumulation point x^* must be a KKT point for $(\tilde{P}_{\bar{c}})$ and that $d^0(x^*, \bar{c}, H^*) = 0$. As c_k remains fixed after $k = N$, we conclude from Step 3(iii) of Algorithm 1 that $\bar{c} + \min_j \{\bar{\mu}^j(x_k)\} \geq \gamma$, for all $k \geq N$. By continuity (Proposition 2.2), we have

$$\bar{c} + \min_j \{\bar{\mu}^j(x^*)\} \geq \gamma.$$

We may invoke Lemma 3.2 to conclude that $f^j(x^*) = 0$, $j = 1, \dots, m_e$, i.e. $x^* \in \Omega$. Finally, in view of Proposition 2.1, we see that x^* is a KKT point for (P) . \square

4 GENERAL CONVERGENCE ANALYSIS

Consider now the general nonlinear programming problem (P) in the case where $m_i > 0$. If Algorithm 1 generates a finite sequence terminating at x_N , the stopping criterion and feasibility properties of the algorithm guarantee that x_N is feasible for (P) , and $d_N^0 = 0$ implies that x_N is a KKT point for (\tilde{P}_{c_N}) . Hence, Proposition 2.1 implies that x_N is a KKT point for (P) . We state this as a proposition.

PROPOSITION 4.1. *Suppose Algorithm 1 generates a finite sequence with final iterate x_N . Then x_N is a KKT point for (P) .*

Now consider the case in which Algorithm 1 generates an infinite sequence. As a first step toward proving convergence to KKT points, we establish an extension of Lemma 3.2 (this result was informally invoked in Section 2).

LEMMA 4.2. *Given $x \in \tilde{\Omega}$, $H = H^T > 0$, let $\bar{\mu} = \bar{\mu}(x, \lambda) \in \mathbb{R}^{m_e}$ be the solution of the least squares problem*

$$\min_{\bar{\mu} \in \mathbb{R}^{m_e}} \left\| \nabla f_0(x) + \sum_{j=1}^{m_i} \lambda^j \nabla g_j(x) + \sum_{j=1}^{m_e} \bar{\mu}^j \nabla f_j(x) \right\|^2, \quad (4)$$

where λ^j is the multiplier associated with the linearization of the inequality constraint $g_j(x)$ in the computation of $d^0(x, c, H)$ via (1). If $c > 0$ is such that $c + \bar{\mu}^j > 0$, $j = 1, \dots, m_e$, then $d^0(x, c, H) = 0$ implies $f^j(x) = 0$, $j = 1, \dots, m_e$, i.e. $x \in \Omega$.

Proof. Let $d^0 = d^0(x, c, H) = 0$. The KKT first order necessary conditions of optimality for the QP (1) are as follows: there exists multipliers $\lambda^j \geq 0$, $j =$

$1, \dots, m_i$, and $\psi^j \geq 0$, $j = 1, \dots, m_e$ such that

$$\begin{aligned} Hd^0 + \nabla\phi_c(x) + \sum_{j=1}^{m_e} \psi^j \nabla f_j(x) + \sum_{j=1}^{m_i} \lambda^j \nabla g_j(x) &= 0, \\ f_j(x) + \langle \nabla f_j(x), d^0 \rangle &\leq 0, \quad j = 1, \dots, m_e, \\ g_j(x) + \langle \nabla g_j(x), d^0 \rangle &\leq 0, \quad j = 1, \dots, m_i, \end{aligned}$$

and the complementary slackness conditions

$$\begin{aligned} \psi^j (f_j(x) + \langle \nabla f_j(x), d^0 \rangle) &= 0, \quad j = 1, \dots, m_e, \\ \lambda^j (g_j(x) + \langle \nabla g_j(x), d^0 \rangle) &= 0, \quad j = 1, \dots, m_i, \end{aligned}$$

are satisfied. Substituting the definition of $\phi_c(x)$, and $d^0 = 0$, these conditions reduce to: there exists multipliers $\lambda^j \geq 0$, $j = 1, \dots, m_i$, and $\psi^j \geq 0$, $j = 1, \dots, m_e$ such that

$$\nabla f_0(x) - \sum_{j=1}^{m_e} (c - \psi^j) \nabla f_j(x) + \sum_{j=1}^{m_i} \lambda^j \nabla g_j(x) = 0, \quad (5)$$

$$\begin{aligned} f_j(x) \leq 0, \quad \psi^j f_j(x) &= 0, \quad j = 1, \dots, m_e, \\ g_j(x) \leq 0, \quad \lambda^j g_j(x) &= 0, \quad j = 1, \dots, m_i. \end{aligned} \quad (6)$$

Defining

$$\nu(x) \triangleq \nabla f_0(x) + \sum_{j=1}^{m_i} \lambda^j \nabla g_j(x) + \sum_{j=1}^{m_e} \bar{\mu}^j \nabla f_j(x),$$

we can rewrite (5) as

$$\nu(x) - \sum_{j=1}^{m_e} (\bar{\mu}^j + c - \psi^j) \nabla f_j(x) = 0. \quad (7)$$

Since $\bar{\mu}$ is the solution of the least squares problem (4), the following orthogonality conditions must hold:

$$\langle \nu(x), \nabla f_j(x) \rangle = 0, \quad j = 1, \dots, m_e.$$

In view of (7) and Assumption 2, we immediately conclude that $\nu(x) = 0$ and $\bar{\mu}^j + c - \psi^j = 0$, $j = 1, \dots, m_e$. By assumption, $\bar{\mu}^j + c > 0$, $j = 1, \dots, m_e$, hence we must have $\psi^j > 0$, $j = 1, \dots, m_e$. In view of the complementary slackness conditions in (6), this implies $f_j(x) = 0$, $j = 1, \dots, m_e$. \square

As in Section 3, we now need to establish that, under some assumptions, c_k is increased only finitely often. The argument used for the case without inequality constraints no longer applies as it relied on boundedness of μ_k for bounded x_k , which cannot be invoked here since λ_k is not known to be bounded. In the present case, the second condition in *Step 3(iii)* is crucial.

LEMMA 4.3. *If the infinite sequence $\{x_k\}_{k \in \mathbb{N}}$ generated by Algorithm 1 is bounded, then c_k is increased only finitely often.*

Proof. By contradiction. Suppose c_k is increased infinitely often. Hence, there exists an infinite index set $\mathcal{K} \subseteq \mathbb{N}$ such that $c_{k+1} > c_k$ for all $k \in \mathcal{K}$. Since $\delta > 1$, we see from *Step 3(iii)* of Algorithm 1 that $c_k \rightarrow \infty$ and the following conditions must hold for all $k \in \mathcal{K}$:

$$c_k + \min_j \{\bar{\mu}_k^j\} < \gamma, \quad (8)$$

$$c_k \|H_k d_k^0\| < M. \quad (9)$$

From (8), we conclude that $\bar{\mu}_k$ is unbounded. It then follows from (3) and Assumption 2 that at least one QP multiplier λ_k^j must also be unbounded. On the other hand, from (9) and Assumption 3, we conclude that both $\|H_k d_k^0\| \xrightarrow{k \in \mathcal{K}} 0$ and $\|d_k^0\| \xrightarrow{k \in \mathcal{K}} 0$. The following two equations come directly from the KKT first-order necessary conditions of optimality for the QP (1):

$$H_k d_k^0 + \nabla f_0(x_k) - \sum_{j=1}^{m_e} (c_k - \psi_k^j) \nabla f_j(x_k) + \sum_{j=1}^{m_i} \lambda_k^j \nabla g_j(x_k) = 0, \quad (10)$$

$$\lambda_k^j (g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle) = 0, \quad j = 1, \dots, m_i, \quad (11)$$

where $\lambda_k^j \geq 0$, $j = 1, \dots, m_i$, and $\psi_k^j \geq 0$, $j = 1, \dots, m_e$, are the KKT multipliers. We make the following definition:

$$\alpha_k^j \triangleq \begin{cases} \lambda_k^j & j = 1, \dots, m_i, \\ c_k - \psi_k^{j-m_i} & j = m_i + 1, \dots, m_i + m_e. \end{cases}$$

Clearly, there must exist an infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ and an index $j_0 \in \{1, \dots, m_i + m_e\}$ such that

$$|\alpha_k^{j_0}| \geq |\alpha_k^j|, \quad \forall k \in \mathcal{K}',$$

$j = 1, \dots, m_i + m_e$. Since at least one of the λ_k^j is unbounded, without loss of generality $|\alpha_k^{j_0}| \xrightarrow{k \in \mathcal{K}'} \infty$. Define

$$\zeta_k^j \triangleq \frac{\alpha_k^j}{\alpha_k^{j_0}}, \quad j = 1, \dots, m_i + m_e,$$

for $k \in \mathcal{K}'$. By construction $|\zeta_k^j| \leq 1$, $j = 1, \dots, m_i + m_e$, for all $k \in \mathcal{K}'$, and $|\zeta_k^{j_0}| = 1$ for all $k \in \mathcal{K}'$. Since the sequences of coefficients $\{\zeta_k^j\}_{k \in \mathcal{K}'}$, $j = 1, \dots, m_i + m_e$, are bounded, and the sequence $\{x_k\}_{k \in \mathcal{K}'}$ is bounded by assumption, there must exist an infinite index set $\mathcal{K}'' \subseteq \mathcal{K}'$ and vectors $x^* \in \mathbb{R}^n$, $\zeta^* \in \mathbb{R}^{m_i + m_e}$ such that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}''}} x_k = x^*,$$

and

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}''}} \zeta_k^j = \zeta^{*,j}, \quad j = 1, \dots, m_i + m_e,$$

i.e., the sequences have accumulation points. Boundedness of $\{x_k\}_{k \in \mathbb{N}}$, and our continuity assumptions, imply that $\nabla f_j(x_k)$, $j = 0, 1, \dots, m_e$, and $\nabla g_j(x_k)$, $j = 1, \dots, m_i$, are bounded. Hence, since $\{H_k d_k^0\}_{k \in \mathbb{N}}$ is bounded, dividing (10) through by $\alpha_k^{j_0}$ and taking the limit as k goes to infinity, $k \in \mathcal{K}''$, yields

$$-\sum_{j=1}^{m_e} \zeta^{*,j+m_i} \nabla f_j(x^*) + \sum_{j=1}^{m_i} \zeta^{*,j} \nabla g_j(x^*) = 0. \quad (12)$$

Since $\mathcal{K}'' \subseteq \mathcal{K}$, we know that $d_k^0 \xrightarrow{k \in \mathcal{K}''} 0$. Divide (11) through by $\alpha_k^{j_0}$ and take the limit once again, yielding:

$$\zeta^{*,j} g_j(x^*) = 0, \quad j = 1, \dots, m_i.$$

Thus, for any j such that $\zeta^{*,j} \neq 0$, we must have $g_j(x^*) = 0$. Therefore, since $\zeta^{*,j_0} \neq 0$, (12) contradicts Assumption 2. \square

We are now ready to prove our main result.

THEOREM 4.4. *If Algorithm 1 generates a bounded infinite sequence $\{x_k\}_{k \in \mathbb{N}}$, then any accumulation point x^* is a KKT point for (P).*

Proof. In view of Lemma 4.3, as $\{x_k\}_{k \in \mathbb{N}}$ is bounded, c_k is increased only finitely many times. Suppose c_k is eventually fixed at \bar{c} , i.e. $c_k = \bar{c}$ for all $k \geq N$, where N is finite. We know from Proposition 2.3 that an accumulation point x^* must be a KKT point for $(\tilde{P}_{\bar{c}})$. Let the infinite index set $\mathcal{K} \subseteq \mathbb{N}$ be such that $x_k \xrightarrow{k \in \mathcal{K}} x^*$ and $H_k \xrightarrow{k \in \mathcal{K}} H^*$, where $H^* = H^{*T} > 0$ (we know such H^* and \mathcal{K} exist from Assumption 3). In view of Proposition 2.3, x^* is a KKT point for $(\tilde{P}_{\bar{c}})$ and $d^0(x^*, \bar{c}, H^*) = 0$. To complete the proof, we show that $f_j(x^*) = 0$, $j = 1, \dots, m_e$. As c_k remains fixed after $k = N$, we conclude from *Step 3(iii)* of Algorithm 1 that

$$\bar{c} + \min_j \{\bar{\mu}^j(x_{k+1}, \lambda_k)\} \geq \gamma,$$

for all $k \geq N$. Since $H^* > 0$, the QP (1) for $d^0(x^*, \bar{c}, H^*)$ satisfies the second-order sufficiency conditions of optimality. Hence, we may apply a celebrated result due to S.M. Robinson (Theorem 2.1 in [8]) to conclude that λ_k converges to λ^* on \mathcal{K} and

$$d_k^0 \xrightarrow{k \in \mathcal{K}} 0. \quad (13)$$

In view of (2) and *Step 1(iii)* of Algorithm 1, (13) implies that $d_k \xrightarrow{k \in \mathcal{K}} 0$. This, along with the norm condition in *Step 1(iv)*, guarantees that $\tilde{d}_k \xrightarrow{k \in \mathcal{K}} 0$ as well. Since $t_k \leq 1$, it is clear from *Step 3(i)* of Algorithm 1 that $\|x_{k+1} - x_k\| \leq \|d_k\| + \|\tilde{d}_k\|$. Hence, $x_{k+1} - x_k \xrightarrow{k \in \mathcal{K}} 0$, which implies that $x_{k+1} \xrightarrow{k \in \mathcal{K}} x^*$. Using the continuity of $\bar{\mu}(\cdot, \cdot)$ (Proposition 2.2), we see that

$$\bar{c} + \min_j \{\bar{\mu}^j(x^*, \lambda^*)\} \geq \gamma.$$

Finally, we may invoke Lemma 4.2 to conclude that $f_j(x^*) = 0$, $j = 1, \dots, m_e$. Since the algorithm generates iterates that are feasible for (\bar{P}_c) , we are guaranteed that $g_j(x^*) \leq 0$, $j = 1, \dots, m_i$. Now, we simply apply Proposition 2.1 and the proof is complete. \square

If we make the further assumption that $f_j(\cdot)$, $j = 0, 1, \dots, m_e$, and $g_j(\cdot)$, $j = 1, \dots, m_i$, are three times continuously differentiable, then the following result is a direct consequence of Proposition 3.4 and Theorem 3.7 in [7].

THEOREM 4.5. *If some accumulation point x^* of the sequence $\{x_k\}_{k \in \mathbb{N}}$ generated by Algorithm 1 satisfies the second order sufficiency conditions with strict complementary slackness, and the sequence is bounded, then the entire sequence converges to x^* . Moreover, the convergence is two-step superlinear.*

Note that the premise that the sequence $\{x_k\}_{k \in \mathbb{N}}$ be bounded is not as restrictive as it may seem. It can be insured, e.g., by including in (P) simple bounds on the components of x (since in the present context all iterates satisfy the inequality constraints).

5 IMPLEMENTATION AND NUMERICAL RESULTS

Algorithm 1 has been implemented in FSQP/CFSQP (FSQP is a Fortran subroutine, while CFSQP is written in C), a nonlinear programming package generating feasible iterates (see [10, 4]). For the sake of completeness, we include here the QP's used to compute the directions d^1 and \tilde{d} in the implementations. The strictly feasible descent direction $d^1 = d^1(x, c, d^0)$ is computed as the solution of the QP (inspired by the suggestions in [7]):

$$\begin{aligned} \min_{d^1 \in \mathbb{R}^n, \gamma \in \mathbb{R}} \quad & \frac{\eta}{2} \langle d^0 - d^1, d^0 - d^1 \rangle + \gamma \\ \text{s.t.} \quad & \langle \nabla \phi_c(x), d^1 \rangle \leq \gamma, \\ & f_j(x) + \langle \nabla f_j(x), d^1 \rangle \leq \gamma, \quad j = 1, \dots, m_e, \\ & g_j(x) + \langle \nabla g_j(x), d^1 \rangle \leq \gamma, \quad j = 1, \dots, m_i, \end{aligned}$$

where $\eta = 0.1$. The coefficient $\rho = \rho(d^0)$ is defined as

$$\rho(d^0) \triangleq \frac{\|d^0\|^\kappa}{\|d^0\|^\kappa + \nu},$$

where $\nu = \max\{1/2, \|d^1\|^\tau\}$, $\kappa = 2.1$, and $\tau = 2.5$. Hence, the condition (2) is satisfied. Finally, the Maratos correction $\tilde{d} = \tilde{d}(x, c, d, H)$ is computed as the

solution of the QP (again, inspired by [7]):

$$\begin{aligned} \min_{\tilde{d} \in \mathbb{R}^n} \quad & \frac{1}{2} \langle d + \tilde{d}, H(d + \tilde{d}) \rangle + \langle \nabla \phi_c(x), d + \tilde{d} \rangle \\ \text{s.t.} \quad & f_j(x + d) + \langle \nabla f_j(x), \tilde{d} \rangle \leq -\min\{0.01\|d\|, \|d\|^\tau\}, \quad j = 1, \dots, m_e, \\ & g_j(x + d) + \langle \nabla g_j(x), \tilde{d} \rangle \leq -\min\{0.01\|d\|, \|d\|^\tau\}, \quad j = 1, \dots, m_i. \end{aligned}$$

For scaling purposes, the FSQP/CFSQP implementations actually assign a different penalty parameter to each nonlinear equality constraint. The penalty parameter update is then as follows

$$c_{k+1}^j = \begin{cases} c_k^j & \text{if } c_k^j + \bar{\mu}_k^j \geq \gamma \text{ or } c_k^j \|H_k d_k^0\| \geq M, \\ \max\{\gamma - \bar{\mu}_k^j, \delta c_k^j\} & \text{else.} \end{cases}$$

for $j = 1, \dots, m_e$. The analysis in Sections 3 and 4 still holds with little modification when this update scheme is used. Linear (affine) equality constraints do not have an associated penalty parameter and are not included in $\phi_c(x)$ (or the calculation of SCV below) since the QPs automatically generate directions that are feasible for these constraints. Therefore, the line search guarantees that all iterates satisfy all linear (affine) equality constraints. The values of the various algorithm parameters used for the test problems are as follows: $\alpha = 0.1$, $\beta = 0.5$, $\delta = 2$, $\gamma = 1$, and $M = 10$.

Table 1 lists the results obtained on test problems taken from [3] and [9]. For purposes of comparison, also listed are the results obtained when the test problems were run using VF02AD from the Harwell subroutine library [5], modified so that the stopping criterion was the same as that in CFSQP (see below). All computations were performed on a Sun 4/SPARCstation IPC in double precision and the gradients were computed analytically. In Table 1, # indicates the problem number as listed in [3, 9] and the second column (A) indicates the algorithm used to solve the problem (C for Algorithm 1 as implemented in CFSQP and V for VF02AD). n , m_e , and m_i are as defined in Section 1. NF and NC indicate the number of objective function evaluations and scalar constraint evaluations, respectively. IT is the number of iterations that were required to meet the stopping criterion. $f(x^*)$ is the value of the objective function at the final iterate, and $\|d_*^0\|$ is the norm of the SQP direction at the final iterate. The stopping criterion for the implementation requires that this number be smaller than some user-supplied $\epsilon > 0$, which was equal to 1.E-4 for all problems except #46, where it was 5.E-3, and #27, where it was 1.E-3 (increased due to slow convergence). SCV is defined as

$$\text{SCV} = \sum_{j=1}^{m_e} |f_j(x^*)|,$$

i.e. the sum of the absolute values of the equality constraint violations. The second half of the stopping criterion requires that this value be smaller than some user-supplied $\epsilon_e > 0$, which was set to approximately 1.E-4 for all problems. Finally,

an N in column FE indicates that the given initial point was not feasible for the original problem, while a Y indicates that it was.

Overall the results in Table 1 are favorable, i.e. CFSQP appears to be competitive. VF02AD is also an implementation of an SQP algorithm, but it is *not* a feasible direction algorithm. One would clearly expect the algorithm requiring feasibility to be at some disadvantage over an algorithm not requiring this property. On average, VF02AD does require fewer iterations and function evaluations, but not significantly fewer (with the exception of the constraint function evaluations, which we expect to be larger because of the feasibility requirements in CFSQP). Further, CFSQP must first generate a feasible initial point if such a point is not provided. This, of course, further runs up the function evaluations.

6 CONCLUSION

We have presented and analyzed a modification of a scheme originally proposed by Mayne and Polak for handling nonlinear equality constraints in feasible direction algorithms. We showed how to efficiently incorporate the scheme into a second-order algorithm, making use of the available multiplier estimates from the computation of the SQP direction. The primary advantage of adapting the Mayne-Polak scheme is that with little extra effort we obtain an *exact* differentiable penalty function. This avoids the numerical problems involved with having to increase a penalty parameter without bound. Finally, we saw that in an implementation the algorithm is competitive with a popular algorithm that does not require feasibility.

REFERENCES

1. J. F. Bonnans, E. R. Panier, A.L. Tits, and J. L. Zhou. Avoiding the Maratos effect by means of a nonmonotone line search II. Inequality constrained problems – feasible iterates. *SIAM J. Numer. Anal.*, 29(4):1187–1202, August 1992.
2. J. Hershkovits. A two-stage feasible directions algorithm for nonlinear constrained optimization. *Mathematical Programming*, 36:19–38, 1986.
3. W. Hock and K. Schittkowski. *Test Examples For Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems No. 187*. Springer-Verlag, Berlin, 1981.
4. C. T. Lawrence, J. L. Zhou, and A. L. Tits. *User's Guide for CFSQP Version 2.1: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints*, 1994. ISR TR-94-16r1, Institute for Systems Research, University of Maryland (College Park, MD).
5. Harwell Subroutine Library. *Library Reference Manual*. Harwell, England, 1985.
6. D. Q. Mayne and E. Polak. Feasible direction algorithms for optimization problems with equality and inequality constraints. *Mathematical Programming*, 11:67–80, 1976.
7. E. R. Panier and A. L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Mathematical Programming*, 59:261–276, 1993.
8. S. M. Robinson. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms. *Mathematical Programming*, 7:1–16, 1974.
9. K. Schittkowski. *More Test Examples For Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems No. 282*. Springer-Verlag, Berlin, 1987.

10. J. L. Zhou and A. L. Tits. *User's Guide for FSQP Version 3.3b: A FORTRAN Code for Solving Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints*, 1993. ISR TR-92-107r3, Institute for Systems Research, University of Maryland (College Park, MD).

#	A	n	m_e	m_i	NF	NC	IT	$f(x^*)$	$\ d_*^0\ $	SCV	FE
6	C	2	1	0	14	25	9	.510673485E-12	.15E-05	.93E-09	N
	V				11	11	10	.559226602E-21	.44E-10	.19E-11	
7	C	2	1	0	21	32	10	-.173205081E+01	.32E-07	.54E-10	N
	V				12	12	10	-.173205081E+01	.11E-08	.31E-09	
26	C	3	1	0	38	79	31	.258774381E-13	.85E-06	.11E-09	Y
	V				31	31	29	.467661451E-13	.99E-04	.51E-07	
27	C	3	1	0	40	61	19	.400000055E-01	.19E-04	.87E-08	N
	V				24	24	20	.399969957E-01	.76E-03	.83E-04	
39	C	4	2	0	13	58	13	-.999999999E+00	.33E-04	.42E-09	N
	V				12	24	11	-.100000289E+01	.58E-04	.29E-05	
40	C	4	3	0	5	27	5	-.250000002E+00	.14E-05	.97E-08	N
	V				5	15	5	-.250000719E+00	.21E-04	.15E-05	
42	C	4	1	0	9	15	6	.138578644E+02	.59E-06	.97E-09	N
	V				10	10	7	.138578644E+02	.15E-05	.97E-10	
46	C	5	2	0	28	102	18	.183175245E-06	.25E-04	.10E-09	Y
	V				19	38	17	.835801102E-07	.23E-04	.62E-06	
47	C	5	3	0	20	141	19	-.599325805E-09	.66E-04	.25E-06	Y
	V				33	99	27	.303508081E-11	.89E-04	.11E-07	
60	C	3	1	0	8	18	8	.325682030E-01	.37E-04	.78E-07	N
	V				9	9	7	.325682327E-01	.22E-04	.30E-05	
63	C	3	1	0	8	17	8	.961715172E+03	.27E-06	.61E-10	N
	V				8	8	7	.961715165E+03	.11E-04	.59E-05	
71	C	4	1	1	7	25	7	.170140173E+02	.98E-07	.22E-10	Y
	V				5	10	5	.170140173E+02	.59E-05	.82E-07	
74	C	4	3	2	26	144	22	.512649811E+04	.47E-05	.24E-09	N
	V				12	60	12	.512649811E+04	.60E-08	.14E-09	
75	C	4	3	2	24	136	21	.517441269E+04	.29E-07	.15E-10	N
	V				10	50	10	.517441270E+04	.21E-11	.57E-11	
77	C	5	2	0	17	73	17	.241505129E+00	.18E-04	.24E-09	N
	V				16	32	15	.241505125E+00	.26E-04	.68E-07	
78	C	5	3	0	9	67	9	-.291970045E+01	.63E-05	.96E-07	N
	V				9	27	8	-.291970041E+01	.41E-06	.29E-10	
79	C	5	3	0	12	71	12	.787768210E-01	.80E-05	.36E-09	N
	V				11	33	10	.787768210E-01	.61E-05	.12E-07	
80	C	5	3	0	17	78	10	.539498478E-01	.14E-06	.48E-11	N
	V				7	21	7	.539498474E-01	.91E-07	.11E-07	
217	C	2	1	1	7	26	10	-.800000000E+00	.54E-08	.61E-08	N
	V				8	16	8	-.800000134E+00	.30E-05	.53E-05	
248	C	3	1	1	19	52	11	-.800000000E+00	.18E-06	.46E-09	Y
	V				16	32	13	-.800000000E+00	.82E-07	.98E-08	
263	C	4	2	2	15	150	23	-.999999999E+00	.82E-04	.25E-07	N
	V				18	72	17	-.100000000E+01	.21E-05	.30E-08	
325	C	2	1	2	7	25	6	.379134145E+01	.40E-08	.15E-07	Y
	V				9	27	7	.379134144E+01	.82E-07	.30E-06	

TABLE 1: Results for Test Problems with Algorithm 1 (CFSQP) and VF02AD