Improved Feasible SQP Algorithm for Nonlinear Programs with Equality Constrained Sub-Problems

Zhijun Luo[1], Guohua Chen[3] and Simei Luo[4]
Department of Mathematics & Applied Mathematics, Hunan University of Humanities, Science and Technology, Loudi, China
Email: ldlzj123@163.com

Zhibin Zhu[2]
School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin, China
Email: zhu_zhibin@163.com

***Abstract*—This paper proposed an improved feasible sequential quadratic programming (FSQP) method for nonlinear programs. As compared with the existing SQP methods which required solving the QP sub-problem with inequality constraints in single iteration, in order to obtain the feasible direction, the method of this paper is only necessary to solve an equality constrained quadratic programming sub-problems. Combined the generalized projection technique, a height-order correction direction is yielded by explicit formulas, which can avoids Maratos effect. Furthermore, under some mild assumptions, the algorithm is globally convergent and its rate of convergence is one-step superlinearly. Numerical results reported show that the algorithm in this paper is effective.**

***Index Terms*—Nonlinear programs, FSQP method, Equality constrained quadratic programming, Global convergence, Superlinear convergence rate**

I. Introduction

Consider the following nonlinear programs

$$\begin{aligned} &\min\ f(x) \\ &s.t.\quad g_j(x) \le 0,\ j \in I = \{1, 2, \cdots, m\}, \end{aligned} \tag{1}$$

Where $f(x),\ g_j(x): R^n \to R (j \in I)$ are continuously differentiable functions. Denote the feasible set for (1) by

$$X = \{x \in R^n \mid g_j(x) \le 0,\ j \in I\}.$$

The Lagrangian function associated with (1) is defined as follows:

$$L(x,\ \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

A point $x \in X$ is said to be a KKT point of (1), if it is satisfies the equalities

$$\nabla f(x) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x) = 0,$$

$$\lambda_j g_j(x) = 0,\ j \in I,$$

where $\lambda = (\lambda_1, \cdots, \lambda_m)^T$ is nonnegative, and λ is said to be the corresponding KKT multiplier vector.

Method of Sequential Quadratic Programming (SQP) is an important method for solving nonlinearly constrained optimization [1, 2, 18]. It generates iteratively the main search direction d_0 by solving the following quadratic programming (QP) sub-problem:

$$\begin{aligned} &\min\ \nabla f(x)^T d + \frac{1}{2} d^T H d \\ &s.t.\quad g_j(x) + \nabla g_j(x)^T d \le 0,\ j \in I, \end{aligned} \tag{2}$$

where $H \in R^{n \times n}$ is a symmetric positive definite matrix. However, such type SQP algorithms have two serious shortcomings:

1) SQP algorithms require that the relate QP sub-problems (2) must be consistency;

2) There exists Matatos effect.

Many efforts have been made to overcome the shortcomings through modifying the quadratic sub-problem (2) and the direction d [4, 5, 7, 8]. Some algorithms solve the problem (1) by using the idea of filter method or trust-region [13, 16, 17].

For the problem (2), it is also a hot topic to solve the QP problem like (2) in the field of optimization. By using the idea of active constraints set, some algorithms solve step by step a series of corresponding QP problems with only equality constraints to obtain the optimum solution to the QP sub-problem (2). P. Spellucci [6] proposed a new method, the d_0 is obtained by solving QP sub-problem with only equality constraints:

This work was supported in part by the National Natural Science Foundation (11061011) of China, and the Educational Reform Research Fund of Hunan University of Humanities, Science and Technology (NO.RKJGY1030), corresponding author, E-mail: ldlzj123@163.com .

doi:10.4304/jcp.8.6.1496-1503

$$\begin{aligned} &\min \ \nabla f(x)^T d + \frac{1}{2} d^T H d \\ &s.t. \ \ g_j(x) + \nabla g_j(x)^T d = 0, \ j \in A \subseteq I, \end{aligned} \tag{3}$$

where the so-called working set $A \subseteq I$ is suitably determined. If $d_0 = 0$ and $\lambda \ge 0$ (λ is said to be the corresponding KKT multiplier vector.), the algorithm stops. The most advantage of these algorithms is merely necessary to solve QP sub-problems with only equality constraints. However, if $d_0 = 0$, but $\lambda < 0$, the algorithm will not implement successfully. In [10], proposed an SQP method for general constrained optimization. Firstly, make use of the technique which handle the general constrained optimization as an inequality parametric programming, then, consider a new quadratic programming with only equality constraints as follow:

$$\begin{aligned} &\min \ \nabla f(x)^T d + \frac{1}{2} d^T H d \\ &s.t. \ \ g_j(x) + \nabla g_j(x)^T d = -\min\{0, \ \pi(x)\}, \ \ j \in J(x). \end{aligned}$$

Where $\pi(x)$ is a suitable vector, $J(x)$ is a suitable approximate active set. But the QP problems may no solution under some conditions. Recently, Zhu [14] Consider the following QP sub-problem:

$$\begin{aligned} &\min \ \nabla f(x)^T d + \frac{1}{2} d^T H d \\ &s.t. \ \ p_j(x) + \nabla g_j(x)^T d = 0, \ j \in L. \end{aligned} \tag{4}$$

where $p_j(x)$ is a suitable vector, L is a suitable approximate active, which guarantees to hold that if $d_0 = 0$, then x is a KKT point of (1), i.e., if $d_0 = 0$, then it holds that $\lambda \ge 0$. Depended strictly on the strict complementarity, which is rather strong and difficult for testing, the superlinear convergence properties of the SQP algorithm are obtained. For avoiding the superlinear convergence depend strictly on the strict complementarity,

Another some SQP algorithms (see [15]) have been proposed, however it is regretful that these algorithms are infeasible SQP type and nonmonotone. In [16], a feasible SQP algorithm is proposed. Using generalized projection technique, the superlinear convergence properties are still obtained under weaker conditions without the strict complementarity.

We will develop an improved feasible SQP method for solving optimization problems based on the one in [14]. The traditional FSQP algorithms, in order to prevent iterates from leaving the feasible set, and avoid Maratos effect, it needs to solve two or three QP sub-problems like (2). In our algorithm, per single iteration, it is only necessary to solve an equality constrained quadratic programming, which is very similar to (4). Obviously, it is simpler to solve the equality constrained QP problem than to solve the QP problem with inequality constraints. In order to void the Maratos effect, combined the generalized projection technique, a height-order correction direction is computed by an explicit formula, and it plays an important role in avoiding the strict complementarity. Furthermore, its global and superlinear convergence rate is obtained under some suitable conditions.

This paper is organized as follows: In Section II, we state the algorithm; the well-defined of our approach is also discussed, the accountability of which allows us to present global convergence guarantees under common conditions in Section III, while in Section IV we deal with superlinear convergence. Finally, in Section V, numerical experiments are implemented.

II. Description of Algorithm

The active constraints set of (1) is denoted as follows:

$$I(x) = \{ j \in I \mid g_j(x) = 0, \ j \in I \}. \tag{5}$$

Now, the following algorithm is proposed for solving the problem (1).

Algorithm A:

Step 0 Initialization:

Given a starting point $x^0 \in X$, and an initial symmetric positive definite matrix $H_0 \in R^{n \times n}$. Choose parameters $\varepsilon_0 \in (0,1), \ \alpha \in (0, \frac{1}{2}), \ \tau \in (2,3)$. Set $k = 0$;

Step 1. Computation of an approximate active set J_k .

Step 1.1. For the current point $x^k \in X$, set $i = 0$,

$$\varepsilon_i(x^k) = \varepsilon_0 \in (0,1) .$$

Step 1.2. If $\det(A_i(x^k)^T A_i(x^k)) \ge \varepsilon_i(x^k)$, let $J_k = J(x^k), \ A_k = A(x^k), \ i(x^k) = i$, and go to Step 2. Otherwise go to Step 1.3, where

$$\begin{aligned} &J_i(x^k) = \{ j \in I \mid -\varepsilon_i(x^k) \le g_j(x^k) \le 0 \}, \\ &A_i(x^k) = (\nabla g_i(x^k), \ j \in J_i(x^k)). \end{aligned} \tag{6}$$

Step 1.3. Let $i = i + 1, \ \varepsilon_i(x^k) = \frac{1}{2} \varepsilon_{i-1}(x^k)$, and go to Step1. 2.

Step 2. Computation of the vector d_0^k .

Step 2.1

$$\begin{aligned} &B_k = (A_k^T A_k)^{-1} A_k^T, v^k = (v_j^k, j \in J_k) = -B_k \nabla f(x^k), \\ &p_j^k = \begin{cases} -v_j^k, v_j^k < 0 \\ g_j(x^k), v_j^k \ge 0 \end{cases} \qquad p^k = (p_j^k, j \in J_k). \end{aligned} \tag{7}$$

Step 2.2 Solve the following equality constrained QP Sub problem at x^k :

$$\begin{aligned} &\min \ \nabla f(x^k)^T d + \frac{1}{2} d^T H_k d \\ &s.t. \ \ p_j^k + \nabla g_j(x^k)^T d = 0, \ j \in J_k. \end{aligned} \tag{8}$$

Let d_0^k be the KKT point of (8), and $b^k = (b_j^k, j \in J_k)$ be the corresponding multiplier vector.

If $d_0^k = 0$, STOP. Otherwise, CONTINUE;

Step 3

Computation of the feasible direction with descent d^k:

$$d^k = d_0^k - \delta_k A_k (A_k^T A_k)^{-1} e_k \tag{9}$$

Where $e_k = (1, \cdots, 1)^T \in R^{|J_k|}$, and

$$\delta_k = \frac{\| d_0^k \| (d_0^k)^T H_k d_0^k}{2 | \mu^{kT} e_k | \| d_0^k \| + 1}, \ \mu^k = -(A_k^T A_k)^{-1} A_k^T \nabla f(x^k)$$

Step 4. Computation of the high-order revised direction $\tilde{d}^k$:

$$\tilde{d}^k = -A_k (A_k^T A_k)^{-1} (\| d_0^k \|^\tau e_k + \tilde{g}_{J_k}(x^k + d^k)), \tag{10}$$

where

$$\tilde{g}_{J_k}(x^k + d^k) = g_{J_k}(x^k + d^k) - g_{J_k}(x^k) - \nabla g_{J_k}(x^k)^T d^k.$$

Step 5. Line search:

Compute t_k, the first number t in the sequence $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots\}$ satisfying

$$f(x^k + td^k + t^2 \tilde{d}^k) \le f(x^k) + \alpha t \nabla f(x^k)^T d^k, \tag{11}$$

$$g_j(x^k + td^k + t^2 \tilde{d}^k) \le 0, j \in I. \tag{12}$$

Step 6. Update:

Obtain H_{k+1} by updating the positive definite matrix H_k using some quasi-Newton formulas. Set $x^{k+1} = x^k + t_k d^k + t^2 \tilde{d}^k$, and $k = k+1$. Go back to step 1.

Throughout this paper, following basic assumptions are assumed.

H2.1 The feasible set $X \ne \Phi$, and functions $f(x)$, $g_j(x), j \in I$ are twice continuously differentiable.

H2.2 $\forall x \in X$, the vectors $\{\nabla g_j(x), j \in I(x)\}$ are linearly independent.

Lemma 2.1 Suppose that H2.1and H2.2 hold, then

1) For any iteration, there is no infinite cycle in step 1.

2) If a sequence $\{x^k\}$ of points has an accumulation point, then there exists a constant $\bar{\varepsilon} > 0$ such that $\varepsilon_{k,ik} > \bar{\varepsilon}$ for k large enough.

Proof.

1) Suppose that the desired conclusion is false, that is to say, there exists some *k*, such that there is an infinite cycle in Step 1, then we obtain, $\forall i = 1, 2, \cdots$, that $A_{k,i}$ is not of full rank, i.e., it holds that

$$\det(A_{k,i}^T A_{k,i}) = 0, \ i = 1, 2, \cdots, \tag{13}$$

And by (6), we can know that $J_{k,i+1} \subseteq J_{k,i}$. Since there are only finitely many choices for $J_{k,i} \subseteq I$, it is sure that $J_{k,i+1} \equiv J_{k,i} \triangleq \tilde{L}_k$ for i large enough. From (6) and (13), with $i \to \infty$, we obtain

$$\tilde{L}_k = I(x^k), \ \det(A_{I(x^k)}^T A_{I(x^k)}) = 0.$$

This is a contradiction to H 2.2, which shows that the statement is true.

2) Suppose K is an infinite index set such that $\{x^k\}_{k \in K} \to x^*$. We suppose that the conclusion is false, i.e., there exists $K' \subseteq K(|K'| = \infty)$, such that

$$\varepsilon_{k,i_k} \to 0, \ k \in K', \ k \to \infty.$$

Let $\tilde{L}_k = J_{k,i_k - 1}$. From the definition of ε_{k,i_k}, it holds, for $k \in K'$, k large enough, that

$$\det(A_{\tilde{L}_k}^T A_{\tilde{L}_k}) = 0, \ -2\varepsilon_{k,i_k} \le g_j(x^k) \le 0, \ j \in \tilde{L}_k. \tag{14}$$

Since there are only finitely many choices for sets $\tilde{L}_k \subseteq I$, it is sure that there exists $K'' \subseteq K'(|K''| = \infty)$, such that $\tilde{L}_k \equiv \tilde{L}, (k \in K'')$, for k large enough. Denote $\tilde{A} = \{\nabla g_j(x^*) \mid j \in \tilde{L}\}$, then, let $k \in K'', k \to \infty$, from (14), it holds that

$$\det(\tilde{A}^T \tilde{A}) = 0, \ g_j(x^*) = 0, \ j \in \tilde{L} \subseteq I(x^*).$$

This is a contradiction to H 2.2, too, which shows that the statement is true.

Lemma 2.2 For the QP sub-problem (8) at x^k, if $d_0^k = 0$, then $\boldsymbol{x}^k$ is a KKT point of (1). If $d_0^k \ne 0$, then x^k computed in step 4 is a feasible direction with descent of (1) at x^k.

Proof.

By the KKT conditions of QP sub-problem (8), we have

$$\nabla f(x^k) + H_k d_0^k + A_k b^k = 0,$$
$$\mathrm{p}_j^k + \nabla g_j(x^k)^T d_0^k = 0, \ j \in J_k,$$

If $d_0^k = 0$, we obtain

$$\nabla f(x^k) + A_k b^k = 0, \ \ \mathrm{p}_j^k = 0, \ \ j \in J_k,$$

Thereby, from (7) and $x^k \in X$, $\forall k$ implies that

$$g_j(x^k) = 0, \ v_j^k \ge 0, \ j \in J_k.$$

In addition, we have $b^k = -B_k \nabla f(x^k) = v^k$, in a word, we obtain

$$\nabla f(x^k) + A_k b^k = 0, \ \ g_j(x^k) = 0, \ b_j^k \ge 0, \ \ j \in J_k,$$

Let $b_j^k = 0, j \in I \setminus J_k$, which shows that x^k is a KKT point of (1).

If $d_0^k \ne 0$, we have

$$g_{J_k}(x^k)^T d^k = A_k^T d^k = -p^k - \delta_k e_k,$$

$$\nabla f(x^k)^T d_0^k = -(d_0^k)^T H_k d_0^k + b^{kT} p^k,$$

So,

$$\nabla f(x^k)^T d^k = \nabla f(x^k)^T d_0^k - \delta_k \nabla f(x^k)^T A_k (A_k^T A_k)^{-1} e_k$$
$$\le -\frac{1}{2}(d_0^k)^T H_k d_0^k + b^{kT} p^k \le -\frac{1}{2}(d_0^k)^T H_k d_0^k < 0.$$

Thereby, we know that d^k is a feasible descent direction of (1) at x^k.

III. Global convergence of Algorithm

In this section, firstly, it is shown that Algorithm A given in section 2 is well-defined, that is to say, for every k, that the line search at Step 5 is always successful

Lemma 3.1 The line search in step 5 yields a stepsize $t_k = (\frac{1}{2})^i$ for some finite $i = i(k)$.

Proof.

It is a well-known result according to Lemma 2.2. For (11),

$$s \triangleq f(x^k + td^k + t^2\tilde{d}^k) - f(x^k) - \alpha t \nabla f(x^k)^T d^k$$
$$= \nabla f(x^k)^T (td^k + t^2\tilde{d}^k) + o(t) - \alpha t \nabla f(x^k)^T d^k$$
$$= (1-\alpha)t\nabla f(x^k)^T d^k + o(t).$$

For (12), if

$$j \notin I(x^k),\ g_j(x^k) < 0;$$
$$j \in I(x^k),\ g_j(x^k) = 0,\ \nabla g_j(x^k)^T d^k < 0,$$

so we have

$$g_j(x^k + td^k + t^2\tilde{d}^k) = \nabla f(x^k)^T (td^k + t^2\tilde{d}^k) + o(t)$$
$$= \alpha t \nabla g_j(x^k)^T d^k + O(t).$$

In the sequel, the global convergence of Algorithm A is shown. For this reason, we make the following additional assumption.

H3.1 $\{x^k\}$ is bounded, which is the sequence generated by the algorithm, and there exist constants $b \ge a > 0$, such that $a\|y\|^2 \le y^T H_k y \le b\|y\|^2$, for all k and all $y \in R^n$.

Since there are only finitely many choices for sets $J_k \subseteq I$, and the sequence $\{d_0^k, d_1^k, \tilde{d}^k, v^k, b^k\}$ is bounded, we can assume without loss of generality that there exists a subsequence K, such that

$$x^k \to x^*, H_k \to H_*, d_0^k \to d_0^*, d^k \to d^*, \tilde{d}^k \to \tilde{d}^*,$$
$$b^k \to b^*, v^k \to v^*, J_k \equiv J \ne \Phi, k \in K, \qquad (15)$$

where J is a constant set.

Theorem 3.2 The algorithm either stops at the KKT point x^k of the problem (1) in finite number of steps, or generates an infinite sequence $\{x^k\}$ any accumulation point x^* of which is a KKT point of the problem (1).

Proof.

The first statement is easy to show, since the only stopping point is in step 3. Thus, assume that the algorithm generates an infinite sequence $\{x^k\}$, and (15) holds. According to Lemma 2.2, it is only necessary to prove that $d_0^* = 0$. Suppose by contradiction that $d_0^* \ne 0$. Then, from Lemma 2.2, it is obvious that d^* is well-defined, and it holds that

$$\nabla f(x^*)^T d^* < 0,\ \nabla g_j(x^*)^T d^* < 0,\ j \in I(x^*) \subseteq J \qquad (16)$$

Thus, from (16), it is easy to see that the step-size t_k obtained in step 5 are bounded away from zero on K, *i.e.*

$$t_k \ge t_* = \inf\{t_k,\ k \in K\} > 0,\ k \in K. \qquad (17)$$

In addition, from (11) and Lemma 2.2, it is obvious that $\{f(x^k)\}$ is monotonous decreasing. So, according to assumption H 2.1, the fact that $\{x^k\}_K \to x^*$ implies that

$$f(x^k) \to f(x^*),\ k \to \infty. \qquad (18)$$

So, from (11), (16), (17), it holds that

$$0 = \lim_{x \in K}(f(x^k) - f(x^*)) \le \lim_{x \in K}(\alpha t_k \nabla f(x^k)^T d^k)$$
$$\le \frac{1}{2}\alpha t_* \nabla f(x^*)^T d^* < 0,$$

which is a contradiction thus $\lim_{x \to \infty} d_0^k = 0$. Thus, x^* is a KKT point of (1).

IV. The Rate of Convergence

Now we discuss the convergent rate of the algorithm, and prove that the sequence $\{x^k\}$ generated by the algorithm is one-step super-linearly convergent under some mild conditions without the strict complementarily. For this purpose, we add some regularity hypothesis.

H 4.1 The sequence $\{x^k\}$ generated by Algorithm A is bounded, and possess an accumulation point x^*, such that the KKT pair (x^*, u^*) satisfies the strong second-order sufficiency conditions, i.e.,

$$d^T \nabla_{xx}^2 L(x^*, u^*) d > 0,$$
$$\forall d \in \Omega \triangleq \{d \in R^n : d \ne 0, \nabla g_{I+}(x^*)^T d = 0\},$$
$$L(x,u) = f(x) + \sum_{j \in I} u_j g_j(x),\ I^+ = \{j \in I : u_j^* > 0\}.$$

Lemma 4.1 Suppose that assumptions H 2.1-H 3.1 hold, then,

1) There exists a constant $\zeta > 0$, such that
$$\|(A_k^T A_k)^{-1}\| \le \zeta;$$

2) $\lim_{k \to \infty} d_0^k = 0;\ \lim_{k \to \infty} d^k = 0;\ \lim_{k \to \infty} \tilde{d}^k = 0;$

3)
$$\|d^k\| \sim \|d_0^k\|,\ \|\tilde{d}^k\| = O(\|d^k\|^2),$$
$$\|d^k - d_0^k\| = O(\|d_0^k\|^3),\ \|\tilde{d}^k\| = O(\|d^k\|^2).$$

Proof.

1) By contradiction, suppose that sequence $\{\|(A_k^T A_k)^{-1}\|\}$ is unbounded, then there exists an infinite subset K, such that

$$\|(A_k^T A_k)^{-1}\| \to \infty,\ (k \in K).$$

In view of the boundedness of $\{x^k\}$ and J_k being a subset of the finite set $I=\{1,\ 2,\ \cdots,m\}$ as well as Lemma 2.1, we know that there exists an infinite index set $K^{'}\subseteq K$ such that

$$x^k\to \tilde{x},\ J_k\equiv J^{'},\ \forall k\in K^{'},\ \det(A_k^T A_k)\geq \overline{\varepsilon},\ \varepsilon_k\geq \overline{\varepsilon}.$$

As a result,

$$\lim_{k\in K^{'}}(A_k^T A_k)=\nabla g_{J^{'}}(\tilde{x})^T\nabla g_{J^{'}}(\tilde{x}),$$

$$\det(\nabla g_{J^{'}}(\tilde{x})^T\nabla g_{J^{'}}(\tilde{x}))\geq \overline{\varepsilon}>0.$$

Hence, we obtain $\|(A_k^T A_k)^{-1}\|\to\|\nabla g_{J^{'}}(\tilde{x})^T\nabla g_{J^{'}}(\tilde{x})\|$, this contradict $\|(A_k^T A_k)^{-1}\|\to\infty,\ (k\in K)$. So the first conclusion 1) follows.

2) We firstly show that $\lim_{k\to\infty} d_0^k=0$.

We suppose by contradiction that $\lim_{k\to\infty} d_0^k\neq 0$, then there exist an infinite index set K and a constant $\sigma>0$ such that $\|d_0^k\|>\sigma$ holds for all $k\in K$. Taking notice of the boundedness of $\{x^k\}$, by taking a subsequence if necessary, we may suppose that

$$x^k\to\tilde{x},\ J_k\equiv J^{'},\ \forall k\in K.$$

Using Taylor expansion, we analyze the first search inequality of Step 5, combining the proof of Theorem 3.2, the fact that $x^k\to x^*,(k\to\infty)$ implies that it is true.

The proof of $\lim_{k\to\infty} d^k=0$; $\lim_{k\to\infty}\tilde{d}^k=0$ are elementary from the result of 1) as well as formulas (9) and (10).

3) The proof of 3) is elementary from the formulas (9), (10) and assumption H2.1.

Lemma 4.2. Let H2.1 to H4.1 holds, $\lim_{k\to\infty}\|x^{k+1}-x^k\|=0$. Thereby, the entire sequence $\{x^k\}$ converges to x^* i.e. $x^k\to x^*,k\to\infty$.

Proof.

From the Lemma 4.1, it is easy to see that

$$\lim_{k\to\infty}\|x^{k+1}-x^k\|=\lim_{k\to\infty}(\|t_k d^k+t_k^2\tilde{d}^k\|)$$

$$\leq\lim_{k\to\infty}(\|d^k\|+\|\tilde{d}^k\|)=0$$

Moreover, together with Theorem 1.1.5 in [4], it shows that $x^k\to x^*,k\to\infty$

Lemma 4.3 It holds, for k large enough, that

1) $J_k\equiv I(x^*)\triangleq I_*,b^k\to u_{I_*}=(u_j^*,j\in I_*),v^k\to(u_j^*,j\in I_*)$

2) $I^+\subseteq L_k=\{j\in J_k:g_j(x^k)+\nabla g_j(x^k)^T d_0^k=0\}\subseteq J_k$.

Proof.

1) Prove $J_k\equiv I_*$.

On one hand, from Lemma 2.1, we know, for k large enough, that $I_*\subseteq J_k$. On the other hand, if it doesn't hold that $J_k\subseteq I_*$, then there exist constants j_0 and $\beta>0$, such that

$$g_{j_0}(x^*)\leq-\beta<0,\ j_0\in J_k.$$

So, according to $d_0^k\to 0$ and the functions $g_j(x)$, ($j\in I$) are continuously differentiable, for k large enough, if $v_{j_0}^k<0$, we have

$$p_{j_0}(x^k)+\nabla g_{j_0}(x^*)^T d_0^k=-v_{j_0}^k+\nabla g_{j_0}(x^*)^T d_0^k$$

$$\geq-\frac{1}{2}v_{j_0}^k>0.$$

Otherwise,

$$p_{j_0}(x^k)+\nabla g_{j_0}(x^*)^T d_0^k$$

$$=g_{j_0}(x^k)+\nabla g_{j_0}(x^*)^T d_0^k\leq-\frac{1}{2}\beta<0,\ \ (v_{j_0}^k\geq 0)$$

which is contradictory with (8) and the fact $j_0\in J_k$. So, $J_k\equiv I_*$ (for k large enough).

Prove that $b^k\to u_{I_*}=(u_j^*,j\in I_*),v^k\to(u_j^*,j\in I_*)$.

For the $v^k\to(u_j^*,j\in I_*)$ statement, we have the following results from the definition of v^k,

$$v^k\to--B_*\nabla f(x^*)=-(A_*^T A_*)^{-1}A_*^T\nabla f(x^*)$$

In addition, since x^* is a KKT point of (1), it is evident that

$$\nabla f(x^*)+A_*u_{I_*}=0,\ u_{I_*}=-B_*\nabla f(x^*)$$

i.e. $u_{I_*}=-(A_*^T A_*)^{-1}A_*^T\nabla f(x^*)$.

Otherwise, from (8), the fact that $d_0^k\to 0$ implies that

$\nabla f(x^k)+H_k d_0^k+A_k b^k=0,b^k\to-B_*\nabla f(x^*)=u_{I_*}$.

The claim holds.

2) For $\lim_{k\to\infty}(x^k,d_0^k)=(x^*,0)$, we have $L_k\subseteq I(x^*)$. Furthermore, it has $\lim_{x\to\infty}u_{I+}^k=u_{I+}^*>0$, so the proof is finished.

In order to obtain super-linear convergence, a crucial requirement is that a unit step size is used in a neighborhood of the solution. This can be achieved if the following assumption is satisfied.

H4.2 Let $\|(\nabla_{xx}^2 L(x^k,u_{J_k}^k)-H_k)d^k\|=o(\|d^k\|)$, where $L(x,u_{J_k}^k)=f(x)+\sum_{j\in J_k}u_{J_k}^k g_j(x)$.

Lemma 4.4 Suppose that Assumption H 2.1 to H 4.2 are all satisfied. Then, the step size in Algorithm A always one, i.e. $t_k\equiv 1$, if k is large enough.

Proof.

It is only necessary to prove that

$$f(x^k+d^k+\tilde{d}^k)\leq f(x^k)+\alpha\nabla f(x^k)^T d^k, \tag{19}$$

$$g_j(x^k+d^k+\tilde{d}^k)\leq 0,j\in I. \tag{20}$$

For (12) if $j\in I\setminus I_*$ we have $g_j(x^*)<0$, $(x^k,\ d^k,\ \tilde{d}^k)\to(x^*,\ 0,\ 0)(k\to\infty)$, then, it is easy to obtain $g_j(x^k+d^k+\tilde{d}^k)\leq 0$ holds.

If $j\in I_*$ we have

$$\begin{aligned} g_j(x^k+d^k+\tilde{d}^k) &= g_j(x^k+d^k)+\nabla g_j(x^k+d^k)^T\tilde{d}^k+O(\|\tilde{d}^k\|^2)\\ &= g_j(x^k+d^k)+\nabla g_j(x^k)^T\tilde{d}^k+O(\|d^k\|\|\tilde{d}^k\|)+O(\|\tilde{d}^k\|^2)\\ &= g_j(x^k+d^k)+\nabla g_j(x^k)^T\tilde{d}^k+O(\|d^k\|\|\tilde{d}^k\|). \end{aligned} \tag{21}$$

In addition, from (9) and (10),

$$\nabla g_j(x^k)^T d^k = \nabla g_j(x^k)^T d_0^k - \delta_k,$$

$$\nabla g_j(x^k)^T \tilde{d}^k = -\| d_0^k \|^\tau - g_j(x^k+d^k) + g_j(x^k) + \nabla g_j(x^k)^T d^k,$$

so, for $\tau \in (2,3)$ we have

$$\begin{aligned} &g_j(x^k+d^k+\tilde{d}^k)\\ &= -\| d_0^k \|^\tau + g_j(x^k) + \nabla g_j(x^k)^T d_0^k - \delta_k + O(\| d^k \|\|\tilde{d}^k\|)\\ &\le -\| d_0^k \|^\tau + O(\| d^k \|\|\tilde{d}^k\|) \le 0. \end{aligned}$$

Hence, the second inequalities of (20) hold for $t=1$ and k is sufficiently large.

The next objective is to show the first inequality of (19) holds. From Taylor expansion and taking into account Lemma 4.1 and Lemma 4.3, we have

$$\begin{aligned} s &\triangleq f(x^k+d^k+\tilde{d}^k) - f(x^k) + \alpha\nabla f(x^k)^T d^k\\ &= \nabla f(x^k)^T(d^k+\tilde{d}^k) + \frac{1}{2}(d^k)^T\nabla^2 f(x^k)^T d^k\\ &\quad -\alpha\nabla f(x^k)^T d^k + o(\| d^k \|^2). \end{aligned} \tag{22}$$

On the other hand, from the KKT condition of (8) and the active set L_k defined by Lemma 4.3 one has

$$\begin{aligned} \nabla f(x^k) &= -H_k d_0^k - \sum_{j\in L_k} u_j^k \nabla g_j(x^k)\\ &= -H_k d^k - \sum_{j\in L_k} u_j^k \nabla g_j(x^k) + o(\| d^k \|^2), \end{aligned} \tag{23}$$

So, from (23) and Lemma 4.3, we have

$$\begin{aligned} &\nabla f(x^k)^T d^k\\ &= -(d^k)^T H_k d^k - \sum_{j\in L_k} u_j^k \nabla g_j(x^k)^T d^k + o(\| d^k \|^2) \quad (24)\\ &= -(d^k)^T H_k d^k - \sum_{j\in L_k} u_j^k \nabla g_j(x^k)^T d_0^k + o(\| d^k \|^2) \end{aligned}$$

$$\begin{aligned} &\nabla f(x^k)^T(d^k+\tilde{d}^k)\\ &= -(d^k)^T H_k d^k - \sum_{j\in L_k} u_j^k \nabla g_j(x^k)^T (d^k+\tilde{d}^k) + o(\| d^k \|^2), \end{aligned} \tag{25}$$

Again, from (21) and Taylor expansion, it is clear that

$$\begin{aligned} &o(\| d^k \|^2) = g_j(x^k+d^k+\tilde{d}^k)\\ &= g_j(x^k) + \nabla g_j(x^k)^T(d^k+\tilde{d}^k) + \frac{1}{2}(d^k)^T\nabla^2 g_j(x^k)^T d^k + o(\| d^k \|^2) \end{aligned}$$

where $j \in L_k$, then, we obtain

$$\begin{aligned} &-\sum_{j\in L_k} u_j^k \nabla g_j(x^k)^T (d^k+\tilde{d}^k)\\ &= \sum_{j\in L_k} u_j^k \nabla g_j(x^k)^T\\ &\quad + \frac{1}{2}(d^k)^T\left(\sum_{j\in L_k} u_j^k \nabla^2 g_j(x^k)^T\right) d^k + o(\| d^k \|^2), \end{aligned} \tag{26}$$

From (25) and (26), we have

$$\begin{aligned} &\nabla f(x^k)^T(d^k+\tilde{d}^k)\\ &= -(d^k)^T H_k d^k - \sum_{j\in L_k} u_j^k \nabla g_j(x^k)\\ &\quad + \frac{1}{2}(d^k)^T\left(\sum_{j\in L_k} u_j^k \nabla^2 g_j(x^k)^T\right) d^k + o(\| d^k \|^2) \end{aligned} \tag{27}$$

Substituting (27) and (24) into (22), it holds that

$$\begin{aligned} s &\triangleq (\alpha-\frac{1}{2})(d^k)^T H_k d^k + (1-\alpha)\sum_{j\in L_k} u_j^k g_j(x^k)\\ &\quad + \frac{1}{2}(d^k)^T\left(\nabla^2 f(x^k) + \sum_{j\in L_k} u_j^k \nabla^2 g_j(x^k)^T - H_k\right) d^k + o(\| d^k \|^2)\\ &= (\alpha-\frac{1}{2})(d^k)^T H_k d^k + (1-\alpha)\sum_{j\in L_k} u_j^k g_j(x^k)\\ &\quad + \frac{1}{2}(d^k)^T\left(\nabla^2 L(x^k, u_{J_k}^k) - H_k\right) d^k + o(\| d^k \|^2). \end{aligned}$$

Then, together assumption H 3.1 and H 4.2 as well as $u_j^k g_j(x^k) \le 0$, shows that

$$s \le (\alpha-\frac{1}{2})a\| d^k \|^2 + o(\| d^k \|^2) \le 0.\ (\alpha \in (0,\ \frac{1}{2})).$$

Hence, the inequality of (19) holds.

Furthermore, in a way similar to the proof of Theorem 5.2 in [5] and in [19, Theorem 2.3], we may obtain the following theorem:

Theorem 4.5 Under all above-mentioned assumptions, the algorithm is superlinearly convergent, i.e., the sequence $\{x^k\}$ generated by the algorithm satisfies that

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

Proof.

From Lemma 4.1 and Lemma 4.4, we can know that the sequence $\{x^k\}$ yielded by Algorithm A has the form of

$$\begin{aligned} x^{k+1} &= x^k + d^k + \tilde{d}^k\\ &= x^k + d_0^k + (d^k + \tilde{d}^k - d_0^k)\\ &\triangleq x^k + d_0^k + \hat{d}^k. \end{aligned}$$

where $\hat{d}^k = (d^k + \tilde{d}^k - d_0^k)$ (for k large enough) and $\| \hat{d}^k \| = O(\| d_0^k \|^3)$. Consequently, we can obtain the result together with Ref. [5] and [19].

V. Numerical Results

In this section, we carry out numerical experiments based on the Algorithm A. The code of the proposed algorithm is written by using MATLAB 7.0 and utilized the optimization toolbox. The results show that the algorithm is effective. During the numerical experiments, it is chosen at random some parameters as follows:

$$\varepsilon_0 = 0.5,\ \alpha = 0.25,\ \tau = 2.25,\ H_0 = I,$$

where I is the $n \times n$ unit matrix. H_k is updated by the BFGS formula [2].

$$H_{k+1} = BFGS(H_k, s^k, y^k),$$

Where,

$$s^k = x^{k+1} - x^k,\ y^k = \theta \hat{y}^k + (1-\theta) H_k s^k,$$

$$\hat{y}^k = \nabla f(x^{k+1}) - \nabla f(x^k) + \sum_{j=1}^{m} u_j^k (\nabla g_j(x^{k+1}) - \nabla g_j(x^k)),$$

$$\theta = \begin{cases} 1, & \text{if } \hat{y}^{kT} s^k \ge 0.2 (s^k)^T H_k s^k, \\ \dfrac{0.8 (s^k)^T H_k s^k}{(s^k)^T H_k s^k - \hat{y}^{kT} s^k}, & \text{otherwise.} \end{cases}$$

In the implementation, the stopping criterion of Step 2 is changed to "If $\| d_0^k \| \le 10^{-8}$, STOP."

TABLE I.
THE DETAIL INFORMATION OF NUMERICAL EXPERIMENTS

NO.	n, m	NT	CPU
HS12	2, 1	10	0
HS43	4, 3	17	10
HS66	3, 8	14	0
HS100	7, 4	18	62
HS113	10, 8	45	50

TABLE II.
THE APPROXIMATE OPTIMAL SOLUTION x^* FOR TABLE I

NO.	the approximate optimal solution x^*
HS12	(1.999999999995731, 3.000000000011285) ^{T}
HS43	(0.000000000000000, 1.000000000000000, 2.000000000000000, −1.000000000000000) ^{T}
HS66	(0.184126482757009, 1.202167866986839, 3.327322301935746) ^{T}
HS100	(2.330499372903103, 1.951372372923884, -0.477541392886392, 4.365726233574537, -0.624486970384889, 1.038131018506466, 1.594226711671913) ^{T}
HS113	(2.171996371254668, 2.363682973701174, 8.773925738481299, 5.095984487967813, 0.990654764957730, 1.430573978920189, 1.321644208159091, 9.828725807883636, 8.280091670090108, 8.375926663907775) ^{T}

This algorithm has been tested on some problems from Ref.[20], a feasible initial point is either provided or obtained easily for each problem. The results are summarized in Table 1 to Tabe 4. The columns of this table have the following meanings:

No.: the number of the test problem in [20];
n: the number of variables;
m: the number of inequality constraints;
NT: the number of iterations;
CPU: the total time taken by the process (unit: millisecond) ;
FV: the final value of the objective function.

TABLE III.
THE APPROXIMATE VALUE OF THE DIRECTION d_0^k FOR TABLE I

NO.	n, m	$\| d_0^k \|$
HS12	2, 1	7.329773437334 E-09
HS43	4, 3	5.473511535838 E-09
HS66	3, 8	8.327832675386 E-09
HS100	7, 4	8.595133692328 E-09
HS113	10, 8	6.056765632745 E-09

TABLE IV.
THE FINAL VALUE OF THE OBJECTIVE FUNCTION FOR TABLE I

NO.	FV
HS12	-29.999999999999705
HS43	-44.000000000000000
HS66	0.518163274181542
HS100	6.806300573744022e+002
HS113	24.306209068179822

VI. CONCLUSION

This paper proposed an improved feasible sequential quadratic programming (FSQP) method for nonlinear programs. As compared with the existing SQP methods which required solving the QP sub-problem with inequality constraints in single iteration, in order to obtain the feasible direction, the method of this paper is only necessary to solve an equality constrained quadratic programming sub-problems. Combined the generalized projection technique, a height-order correction direction is yielded by explicit formulas, which can avoids Maratos effect. Furthermore, under some mild assumptions, the algorithm is globally convergent and its rate of convergence is one-step super-linearly. Numerical results reported show that the algorithm in this paper is effective.

ACKNOWLEDGMENT

The author would like to thank the editors, whose constructive comments led to a considerable revision of the original paper.

REFERENCES

[1] S. P. Han. "Superlinearly Convergent Variable Metric Algorithm for General Nonlinear Programming Problems", Mathematical Programming, Berlin, vol.11 (1), pp. 263–282, December 1976.
[2] M. J. D. Powell. "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations", In: Waston, G.A. (ed). Numerical Analysis. Springer, Berlin, pp. 144–157, 1978.
[3] P. T. Boggs, J. W. Tolle. "A Strategy for Global Convergence in a Sequential Quadratic Programming Algorithm", SIAM J. Num. Anal., Philadelphia, vol. 26 (1), pp. 600–623, June 1989.
[4] E. R. Panier, A. L. Tits. "On Combining Feasibility, Descent and Superlinear Convergence in Inequality

Constrained Optimization", Mathematical Programming, Berlin, vol. 59, pp. 261–276, 1993.
[5] J.F. Binnans, G. Launay. "Sequential quadratic programming with penalization the displacement", SIAM J. Optimization, Philadelphia, vol. 5 (4), pp. 796–812, 1995.
[6] P. Spellucci. "An SQP Method for General Nonlinear Programs Using Only Equality Constrained Subproblems", Mathematical Programming, Berlin, vol. 82 (3), pp. 413–448, August 1998.
[7] C. T. Lawarence, and A.L.Tits. "A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm", SIAM J.Optim., Philadelphia, vol. 11, pp. 1092–1118, 2001.
[8] L. Qi, Y. F. Yang. "Globally and Superlinearly Convergent QP-free Algorithm for Nonlinear Constrained Optimization", Journal of Optimization Theory and Applications, Berlin, vol. 113 (2), pp. 297–323, May 2002. (7)
[9] J.B.Jian, C.M.Tang. "An SQP Feasible Descent Algorithm for Nonlinear Inequality Constrained Optimization without Strict Complementarity", An International Journal Computers and Mathematics with application, vol. 49, pp. 223–238, 2005.
[10] Z.J. Luo, Z.B. Zhu, L.R Wang, and A.J Ren. "An SQP algorithm with equality constrained sub-problems for general constrained optimization", International Journal of Pure and Applied Mathematics, vol. 39 (1): pp. 125-138, 2007.
[11] Richard H. Byrd, Frank E. Curtis, and Jorge Nocedal. "Infeasibility Detection and SQP Methods for Nonlinear Optimization", SIAM J. Optim. Vol. 20 (5), pp. 2281-2299, 2010.
[12] A. F. Izmailov and M. V. Solodov. "A Truncated SQP Method Based on Inexact Interior-Point Solutions of Sub-problems", SIAM J. Optim. Vol. 20 (5), pp. 2584-2613, 2010.
[13] C.G. Shen, W.J. Xue and X.D. Chen. "Global convergence of a robust filter SQP algorithm", European Journal of Operational Research, vol. 206 (1), pp. 34–45, 2010.
[14] Z. B. Zhu, W.D. Zhang, and Z.J. Geng. "A feasible SQP method for nonlinear programming", Applied Mathematics and Computation, vol. 215, pp. 3956–3969, 2010.
[15] Z. J. Luo, G. H. Chen, and L. R. Wang, "A Modified Sequence Quadratic Programming Method for Nonlinear Programming, " cso, pp.458-461, 2011 Fourth International Joint Conference on Computational Sciences and Optimization, 2011.
[16] C. Gu, D.T. Zhu. "A non-monotone line search multidimensional filter-SQP method for general nonlinear programming", Numerical Algorithms, vol. 56 (4), pp.537–559, 2011.
[17] C.L. Hao, X.W. Liu. "A trust-region filter-SQP method for mathematical programs with linear complementarity constraints", Journal of Industrial and Management Optimization, vol. 7 (4), pp. 1041–1055, 2011.
[18] C.L. Hao, X.W. Liu. "Global convergence of an SQP algorithm for nonlinear optimization with over determined constraints", Numerical algebra control and optimization, vol. 2 (1), pp. 19–29, 2012.
[19] J.B. Jian. "Two extension models of SQP and SSLE algorithms for optimization and their superlinear and quadratical convergence", Applied Mathematics--A Journal of Chinese Universities, Set. A, vol. 16 (4), pp.435-444, 2001.
[20] W. Hock, K. Schittkowski, Test examples for nonlinear programming codes, Lecture Notes in Economics and Mathematical Systems, vol. 187, Springer, Berlin, 1981.