# A sequential quadratic programming method without a penalty function or a filter for nonlinear equality constrained optimization*

Xinwei Liu[†] and Yaxiang Yuan[‡]

---

**Abstract.** We present a sequential quadratic programming method without using a penalty function or a filter for solving nonlinear equality constrained optimization. In each iteration, the linearized constraints of the quadratic programming are relaxed to satisfy two mild conditions, the step-size is selected such that either the value of the objective function or the measure of the constraint violations is sufficiently reduced. As a result, our method has two nice properties. Firstly, we do not need to assume the boundedness of the iterative sequence; Secondly, we do not need any restoration phase which is necessary for filter methods. We prove that the algorithm will terminate at either an approximate Karush-Kuhn-Tucker point, or an approximate Fritz-John point, or an approximate infeasible stationary point which is an approximate stationary point for minimizing the $\ell_2$ norm of the constraint violations. By controlling the exactness of the linearized constraints and introducing a second-order correction technique, without requiring linear independence constraint qualification, the algorithm is shown to be locally superlinearly convergent. The preliminary numerical results show that the algorithm is robust and efficient when solving some small- and medium-size problems from the CUTE collection.

**Key words:** sequential quadratic programming, penalty function, filter, regularity, global and local convergence analysis.

---

*This work was partially supported by Grants 10971047, 10831006 of National Natural Science Foundation of China, a grant from Hebei Natural Science Foundation of China and CAS grant kjcx-yw-s7.

[†]Department of Applied Mathematics, Hebei University of Technology, Tianjin 300401, China, mathlxw@hebut.edu.cn

[‡]LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, AMSS, CAS, Beijng 100190, China, yyx@lsec.cc.ac.cn

## 1. Introduction

Consider the nonlinear program with general nonlinear equality constraints

$$\min \ f(x) \tag{1.1}$$

$$\text{s.t.} \ h_j(x) = 0, \ j \in \mathcal{E}, \tag{1.2}$$

where $x \in \Re^n$, $\mathcal{E} = \{1, 2, \dots, m\}$ $(m \le n)$, $f$, $h_j$ $(j \in \mathcal{E})$ are twice continuously differentiable real-valued functions defined on $\Re^n$. Sequential quadratic programming (SQP) approach for problem (1.1)–(1.2) is iterative. Suppose that $x^k$ is the current iterate, SQP solves the quadratic programming (QP) problem

$$\min \ (g^k)^T d + (1/2) d^T B_k d \tag{1.3}$$

$$\text{s.t.} \ h_j^k + (a_j^k)^T d = 0, \ j \in \mathcal{E}, \tag{1.4}$$

where $g^k = \nabla f(x^k)$, $h_j^k = h_j(x^k)$, $a_j^k = \nabla h_j(x^k)$, $B_k \in \Re^{n \times n}$ is the Lagrangian Hessian at $x^k$ or its approximation. Let $d^k \in \Re^n$ be a solution of program (1.3)–(1.4). The new iterate is generated by a line search procedure

$$x^{k+1} = x^k + \alpha_k d^k, \tag{1.5}$$

where the step-length $\alpha_k \in (0, 1]$ is generally selected to satisfy a sufficient descent condition on a penalty function with an appropriately picked penalty parameter. Assuming that $a_j^k, j \in \mathcal{E}$, are linearly independent (thus $A_k = [a_1^k \ \ \dots \ \ a_m^k]$ is of full column rank) and $B_k$ is positive definite (possibly in the null space of $A_k^T \in \Re^{m \times n}$), the SQP method is known to be converging to the Karush-Kuhn-Tucker (KKT) point globally. If $B_k$ is selected suitably, the rapid local convergence can be obtained, for example see [22, 26].

It has been observed in various numerical experiments on standard test problems that methods using a penalty function, including SQP methods and interior-point methods, can be affected by the "inappropriate" selection of the penalty parameter. Although many SQP methods and interior-point methods update the penalty parameter adaptively, the selection for initial value of the penalty parameter is often arbitrary and heuristic, sometimes it is not credible and can cause some difficulties in solving the problems numerically.

In order to avoid the practical problems associated with the setting of the penalty parameter, Fletcher and Leyffer [13] introduced the filter technique, a new strategy for globalizing methods for nonlinear programming, and showed numerically that an SQP trust-region algorithm with the technique performs very promising. Fletcher, Leyffer and Toint [14] proved the global convergence of a filter-SQP algorithm. After that, many globally convergent filter methods have been presented, such as Chin and Fletcher [9], Fletcher, Gould, Leyffer, Toint and Wächter [12], Ribeiro, Karas, Gonzaga [24], Ulbrich, Ulbrich and Vicente [28], Wächter and Biegler [29, 30], and so on. Some of them have also been proved to be locally superlinearly convergent.

Very recently, Gould and Toint [17] introduced a new method for solving equality constrained nonlinear optimization problems. The method did not use a penalty function, a barrier or a filter, and it was proved to be globally convergent. Ulbrich and Ulbrich [27] proposed and analyzed a penalty-function-free nonmonotone trust-region method. Numerical experiments on CUTE problems indicated that both methods performed well. Yamashita and Yabe [32] also presented a trust-region SQP method without a penalty function or a filter for solving constrained optimization with nonlinear equality and nonnegativity constraints. These methods used a trust-region procedure to generate new iterates and solved several different subproblems for coping with the nonlinearities of the objective function and the constraints.

We present a sequential quadratic programming algorithm without a penalty function or a filter for solving the equality constrained optimization problem (1.1)–(1.2) in this paper. The algorithm generates new iterates by line search procedures. In each iteration, the linearized constraints of the quadratic programming are relaxed to satisfy two mild conditions, the step-size is selected such that either the value of the objective function or the measure of the constraint violations is sufficiently reduced. As a result, our method has two nice properties. Firstly, we do not need to assume the boundedness of the iterative sequence. Secondly, we do not need any restoration phase which is necessary for filter methods. Under mild assumptions, we prove that the algorithm will find a KKT point, or a Fritz-John (FJ) point of problem (1.1)–(1.2), or an infeasible stationary point of the problem which is a stationary point for minimizing the $\ell_2$ norm of the constraint violations. By controlling the exactness of the linearized constraints and introducing a second-order correction technique, without requiring linear independence constraint qualification, the algorithm is shown to be locally superlinearly convergent. The preliminary numerical results show that the algorithm is robust and efficient when solving a set of small- and medium-size problems from the CUTE collection [3].

The algorithm is not a straight derivation of Gould and Toint [17], Ulbrich and Ulbrich [27], and Yamashita and Yabe [32]. One of our main contributions in this paper is to develop a set of line search procedures only depending on the decrement of either the value of the objective function or the $\ell_2$ measure of the constraint violations, so that the presented SQP is still globally convergent and locally superlinearly convergent. These line search procedures are not restrictive and play a very important role for the global and local convergence analysis, some of them are inspired by Chin and Fletcher [9] and Fletcher, Leyffer and Toint [14]. Since the algorithm does neither use a penalty function nor a filter, it is not necessary to assume the boundedness of the iterative sequence as usual (see Solodov [25] for a recent discussion on this assumption), and instead we suppose that a level set on a measure of constraint residuals is bounded and the objective function is bounded below, which are similar to the assumptions in unconstrained optimization.

This paper is organized as follows. In Section 2, we firstly describe a QP with relaxed linearized constraints. Two mild conditions are presented. After that, we present our algorithm and then show that the algorithm is well-defined. The global convergence results on the algorithm are proved in Section 3. In Section 4, we show that a superlinear/quadratic step can be obtained by controlling the exactness of the linearized constraints. In order to circumvent the so-called Maratos effect, a second-order correction technique is introduced in the algorithm, and then we prove that the full superlinear step will be accepted. The algorithm is implemented in

Section 5, and preliminary numerical results for some small- and medium-size problems from the CUTE collection [3] are reported.

Throughout the paper, the lower letters represent the vectors, and the capital letters represent the matrices. The superscript and subscript of a vector stand for the corresponding iteration and the corresponding component, respectively. The subscript and superscript of a matrix respectively stand for the corresponding iteration and the corresponding column of the matrix. The expression $\alpha_k = O(\beta_k)$ means that there exists a constant $M$ independent of $k$ such that $\alpha_k \le M\beta_k$, and $\alpha_k = o(\beta_k)$ indicates that $\lim_{\beta_k \to 0} \alpha_k/\beta_k = 0$. If it is not specified, $\|\cdot\|$ is the Euclidean norm. For simplicity, we also use the notations: $h(x) = (h_1(x) \quad \dots \quad h_m(x))^T$, $h^k = h(x^k)$, $h^* = h(x^*)$, $h_j^k = h_j(x^k)$, $f^k = f(x^k)$, $f^{k+1} = f(x^{k+1})$, $v^k = v(x^k)$, $v^{k+1} = v(x^{k+1})$, $g^* = \nabla f(x^*)$, $A(x) = [\nabla h_1(x) \quad \dots \nabla h_m(x)]$, $A_* = A(x^*)$, and $h_{\mathcal{J}_k} = (h_j(x),\ j \in \mathcal{J}_k)^T \in \Re^{|\mathcal{J}_k|}$, $A_{\mathcal{J}_k} = [\nabla h_j(x^k),\ j \in \mathcal{J}_k] \in \Re^{n \times |\mathcal{J}_k|}$, $A_{\mathcal{J}^*} = [\nabla h_j(x^*),\ j \in \mathcal{J}^*] \in \Re^{n \times |\mathcal{J}^*|}$, where $\mathcal{J}_k$ and $\mathcal{J}^*$ are two index sets, $|\mathcal{J}_k|$ and $|\mathcal{J}^*|$ are their cardinalities.

## 2. The algorithm

We present our algorithm and prove that it is well-defined in this section.

**2.1. The QP subproblem with relaxed linearized constraints.** It is well known that, even if the solution $x^*$ of the original problem (1.1)–(1.2) is regular (that is, the linear independence constraint qualification (LICQ) and the second-order sufficient conditions (SOSC) hold at the solution), for any iterate $x^k$ far from $x^*$, the vectors $a_j^k, j \in \mathcal{E}$, are still possibly linearly dependent. In the interior-point algorithmic framework, the matrix $[a_j^k, j \in \mathcal{E}]$ may be asymptotically degenerate. The linear dependence may result in that the subproblem (1.3)–(1.4) is infeasible at $x^k$, while the asymptotical degeneracy may lead to the solution of the subproblem being asymptotically too large in norm. These pitfalls can finally cause the failure of many methods for nonlinear programming (see, for example, [7, 22]). So far, there has been already many works such as [1, 4, 5, 8, 19, 20, 33], whose contributions consist in relaxing the constraints of the QP subproblem (1.3)–(1.4). They have been proved to be able to alleviate or remove the symptom, and improve the convergence properties of the SQP and interior-point methods.

Suppose that $x^k$ is the current iterate, Liu and Yuan [21] proposes to solve the quadratic programming subproblem

$$\min \ (g^k)^T d + (1/2) d^T B_k d \tag{2.1}$$

$$\text{s.t. } A_k^T d = A_k^T d_p^k, \tag{2.2}$$

where $d_p^k \in \Re^n$ approximately minimizes $\|h^k + A_k^T d\|$ and satisfies two prescribed conditions:

**Condition (a).** $\|d_p^k\| \le \kappa_1 \|A_k h^k\|$, where $\kappa_1 > 0$ is a constant;

**Condition (b).** If $\|h^k\| \ne 0$, $\|h^k\| - \|h^k + A_k^T d_p^k\| \ge \kappa_2 \|A_k h^k\|^2 / \|h^k\|$, where $\kappa_2 \in (0,1)$ is a constant.

It has been shown by [15, 16, 21] that the program (2.1)–(2.2) has a unique solution provided that $B_k$ is positive definite in the null space of $A_k^T$. Let $d^k$ be the solution. Since $h^k + A_k^T d^k =$

$h^k + A_k^T d_p^k$, then

$$\|h^k + A_k^T d^k\| \leq (1 - \kappa_2 \|A_k h^k\|^2 / \|h^k\|^2) \|h^k\|,$$

which shows that the constraints in (1.4) have been relaxed.

To relax the linearized constraints by a null-space approach is not a new idea, [5, 22, 23] have used similar techniques to improve the performances of trust-region methods for equality constrained optimization. The above conditions can be guaranteed by the following result.

**Lemma 2.1** *Assume $\|h^k\| \neq 0$. Let $d_p^k = -\alpha_k^c A_k h^k$, where $\alpha_k^c = argmin_{\alpha \in (0,1]} \|h^k - \alpha A_k^T A_k h^k\|$. Then there holds*

$$\|h^k\| - \|h^k + A_k^T d_p^k\| \geq (1/2) \min\{1, \theta_k\} \|A_k h^k\|^2 / \|h^k\|, \tag{2.3}$$

*where $\theta_k = \|A_k h^k\|^2 / \|A_k^T A_k h^k\|^2$.*

*Proof.* This result can be proved by a technique similar to Lemma 2.1 of Liu and Yuan [21]. □

Under the assumptions in Section 3, we have that $\|A_k A_k^T\| \leq \kappa$ for some constant $\kappa \geq 1$. Thus, $\theta_k \geq 1/\|A_k A_k^T\| \geq 1/\kappa$. Then **Condition (b)** follows immediately from (2.3). Since $\|d_p^k\| \leq \alpha_k^c \|A_k h^k\| \leq \|A_k h^k\|$, **Condition (a)** holds naturally.

**2.2. Our algorithm.** Let $L(x, \lambda) = f(x) + \lambda^T h(x)$. The KKT conditions of problem (1.1)–(1.2) are as follows:

$$\nabla_x L(x^*, \lambda^*) = g^* + A_* \lambda^* = 0, \tag{2.4}$$

$$h^* = 0, \tag{2.5}$$

where $x^* \in \Re^n$ is a KKT point and $\lambda^* \in \Re^m$ is the associated Lagrangian multiplier vector.

For every $x \in \Re^n$, we define the measure function on the constraint violations

$$v(x) = \|h(x)\|. \tag{2.6}$$

Moreover, let $\phi(x; d) = \|h(x) + A(x)^T d\| - \|h(x)\|$, where $d \in \Re^n$.

We present our algorithm in the following:

**Algorithm 2.2** *(The algorithm for problem (1.1)–(1.2))*

**Given** *the initial point $x^0 \in \Re^n$, a constant $\sigma \in (0, 1/2)$, two small constants $\xi_1 > 0, \xi_2 > 0$, and the tolerance $\epsilon > 0$. Compute $g^0$, $h^0$, $v^0$, $A_0$ and $B_0$. Set $v_{\max}^0 = 0$, $r_0 = 0.9$. Let $k := 0$;*

**While** $\max(\|\nabla_x L(x^k, \lambda^k)\|, \|h^k\|) > \epsilon$ *and* $\|A_k h^k\| > \epsilon \min(\|h^k\|, 1)$;

**Calculate** $d_p^k$ *approximately minimizing $\|h^k + A_k^T d\|$ on $d$ satisfying* **Conditions (a)–(b)**.

**Solve** *the QP subproblem (2.1)–(2.2). Let $d^k$ be the solution.*

**Select** *the step-size $\alpha_k \in (0, 1]$ as large as possible such that either both inequalities*

$$f(x^k + \alpha_k d^k) - f^k \leq \min\{\sigma \alpha_k (g^k)^T d^k, -\xi_1 v(x^k + \alpha_k d^k)\} \tag{2.7}$$

*and*

$$v(x^k + \alpha_k d^k) \le \max\{(r_k+1)/2, 0.95\} v^k_{\max} \quad \textit{if} \quad v^k_{\max} \neq 0 \tag{2.8}$$

*hold or the inequality*

$$v(x^k + \alpha_k d^k) - v^k \le \min\{\sigma \alpha_k \phi(x^k; d^k), -\xi_2 \alpha_k^2 \|d^k\|^2\} \tag{2.9}$$

*is satisfied.*

**Set** $x^{k+1} = x^k + \alpha_k d^k$.

**If** *(2.9) holds at* $x^{k+1}$ *but not* $x^k$, *set* $v^{k+1}_{\max} = v^k$, *else* $v^{k+1}_{\max} = v^k_{\max}$;

**Compute** $g^{k+1}$, $h^{k+1}$, $v^{k+1}$, $A_{k+1}$, *and update* $B_k$ *to* $B_{k+1}$. *If (2.9) holds, calculate* $r_{k+1} = v^{k+1}/v^k$; *otherwise,* $r_{k+1} = r_k$. *Let* $k := k+1$;

**end (while)**

The algorithm will terminate with three cases: (i) $\|\nabla_x L(x^k, \lambda^k)\| \le \epsilon$ and $\|h^k\| \le \epsilon$; (ii) $\|h^k\| < 1$, and $\|A_k h^k\|/\|h^k\| \le \epsilon$; (iii) $\|h^k\| \ge 1$, but $\|A_k h^k\| \le \epsilon$. If $\epsilon$ is small enough, by (2.4)–(2.5), case (i) implies that $x^k$ is an approximate KKT point. The global convergence results in Section 3 will indicate that $x^k$ is an approximate FJ point for case (ii) and an approximate infeasible stationary point for case (iii), respectively.

For convenience of statement, we refer to the iteration as an *h-type iteration* if (2.9) holds, and an *f-type iteration* if (2.7) and (2.8) are satisfied. Thus, the measure of the constraint violations is decreased for an h-type iteration, and the value of the objective function is decreased for an f-type iteration. It should be noted that there are obvious differences on the h-type iteration and the f-type iteration between this paper and previous works [9, 14] since we are using a line search strategy instead of a trust-region strategy.

The condition (2.8) is not restrictive. We do not calculate $v^0_{\max} = v^0$ at the starting point $x^0$ but set $v^0_{\max} = 0$ directly. With this flexibility, even if $v^0$ is sufficiently small, the algorithm may still take a large step as an f-type iteration without the restriction of (2.8). Since $v^k_{\max}$ is only reset as an h-type iteration starts, this suggests that $v^k_{\max}$ is still large enough and can be reduced sufficiently. We should notice that $r_{k+1}$ remains unchanged for f-type iterations. Since an h-type iteration implies that $v^{k+1} < v^k$ by (2.9), one always has $0 \le r_k < 1$. Thus,

$$r_k < (r_k+1)/2 < 1 \quad \text{and} \quad (r_k+1)/2 \ge 1/2. \tag{2.10}$$

Moreover,

$$\max\{(r_k+1)/2, 0.95\} = \begin{cases} (r_k+1)/2, & \text{if} \quad r_k \ge 0.9; \\ 0.95, & \text{otherwise}, \end{cases} \tag{2.11}$$

where 0.95 is a threshold for (2.8) accepting a possible larger step-size.

**2.3. Well-definedness.** By (2.2) and **Condition (b)**, if $A_k^T d^k = 0$, then $A_k h^k = 0$. On the contrary, if $A_k h^k = 0$, by **Condition (a)**, $d_p^k = 0$. It follows from (2.2) that $A_k^T d^k = 0$. Thus, $A_k^T d^k = 0$ if and only if $A_k h^k = 0$.

Since $h_j : \Re^n \to \Re$ ($j \in \mathcal{E}$) is twice continuously differentiable, $v(x)$ is always directionally differentiable (for example, see Nocedal and Wright [22]). Moreover, we have the following results:

**Lemma 2.3** *Let $v^{'}(x^k; d^k)$ be the directional derivative along $d^k$ at $x^k$. There holds*

$$-\|A_k^T d^k\| \le v^{'}(x^k; d^k) \le \phi(x^k; d^k) \le -\kappa_2 \|A_k h^k\|^2 / \|h^k\|, \tag{2.12}$$

*where $\kappa_2 \in (0, 1)$ is a constant defined in* **Condition (b)**. *Thus, if $\|A_k h^k\| \neq 0$, then $d^k$ is a descent direction of $v(x)$ at $x^k$.*

*Proof.* Since $h(x)$ is twice continuously differentiable on $\Re^n$, by Taylor's Theorem, for $\alpha$ small enough, there exists a $\theta \in (0, \alpha)$ such that

$$\begin{aligned} v(x^k + \alpha d^k) - v^k &= \|h^k + \alpha A_k^T d^k + (1/2)\alpha^2 \sum_{j=1}^m (d^k)^T \nabla^2 h_j(x^k + \theta d^k) d^k\| - \|h^k\| \\ &= \|h^k + \alpha A_k^T d^k\| - \|h^k\| + O(\alpha^2) \\ &\le \alpha(\|h^k + A_k^T d^k\| - \|h^k\|) + O(\alpha^2), \end{aligned} \tag{2.13}$$

while the first equality of (2.13) implies that

$$v(x^k + \alpha d^k) - v^k \ge -\alpha \|A_k^T d^k\| - (1/2)\alpha^2 \sum_{j=1}^m \|(d^k)^T \nabla^2 h_j(x^k + \theta d^k) d^k\|. \tag{2.14}$$

Because

$$v^{'}(x^k; d^k) = \lim_{\alpha \downarrow 0} (v(x^k + \alpha d^k) - v^k)/\alpha,$$

we complete the proof by (2.13), (2.14) and **Condition (b)**. □

**Lemma 2.4** *If $A_k h^k = 0$ and there exists a positive constant $\gamma$ such that $d^T B_k d \ge \gamma \|d\|^2$ for all $d \in \{d : A_k^T d = 0\}$, then*

$$(g^k)^T d^k \le -\gamma \|d^k\|^2. \tag{2.15}$$

*Proof.* If $A_k h^k = 0$, then $A_k^T d^k = 0$. Since $d^k$ satisfies equation

$$g^k + B_k d^k + A_k \lambda^k = 0, \tag{2.16}$$

where $\lambda^k \in \Re^m$ is the multiplier vector, left-multiplying $(d^k)^T$ on the two sides of (2.16), one has

$$(g^k)^T d^k + (d^k)^T B_k d^k = 0.$$

Thus, (2.15) follows from the fact that $(d^k)^T B_k d^k \ge \gamma \|d^k\|^2$ since $d^T B_k d \ge \gamma \|d\|^2$ for all $d \in \{d : A_k^T d = 0\}$. □

**Lemma 2.5** *Suppose that $x^k$ is the current iterate and that the algorithm does not terminate at $x^k$. If there is a constant $\gamma > 0$ such that $d^T B_k d \geq \gamma \|d\|^2$ for all $d \in \{d : A_k^T d = 0\}$, then there always exists a scalar $t_k \in (0, 1]$ such that either (2.9) holds or both (2.7) and (2.8) are satisfied for all $\alpha \in (0, t_k]$. Therefore, Algorithm 2.2 is well-defined.*

*Proof.* Since the algorithm does not terminate at $x^k$, then we have either $\|A_k h^k\| \neq 0$ or the case where $\|A_k h^k\| = 0$, $v^k = 0$ and $\|d^k\| \neq 0$ (that is, $x^k$ is not a KKT point). In what follows we discuss these two cases respectively.

**Case 1.** $\|A_k h^k\| \neq 0$. Then $v^k \neq 0$. It follows from (2.12) that $\phi(x^k; d^k) < 0$. Thus, there exists a constant $\hat{t}_k \in (0, 1]$ such that

$$\min\{\sigma\alpha\phi(x^k; d^k), -\xi_2\alpha^2\|d^k\|^2\} = \sigma\alpha\phi(x^k; d^k) \tag{2.17}$$

for all $\alpha \in (0, \hat{t}_k]$. Due to

$$\begin{aligned} v(x^k + \alpha d^k) &= v^k + \alpha v^{'}(x^k; d^k) + o(\alpha) \\ &\leq v^k + \alpha\phi(x^k; d^k) + o(\alpha), \end{aligned} \tag{2.18}$$

$v(x^k + \alpha d^k) - v^k - \sigma\alpha\phi(x^k; d^k) \leq (1 - \sigma)\alpha\phi(x^k; d^k) + o(\alpha) \leq 0$ for all sufficiently small $\alpha > 0$. Hence, by (2.17), there exists a scalar $t_k \in (0, \hat{t}_k]$ such that (2.9) holds for all $\alpha \in (0, t_k]$.

**Case 2.** $\|A_k h^k\| = 0$, $v^k = 0$ and $\|d^k\| \neq 0$. In this case, $\|A_k^T d^k\| = 0$ and $\phi(x^k; d^k) = 0$. Thus, by (2.12), $v^{'}(x^k; d^k) = 0$. It is obtained from (2.18) that

$$v(x^k + \alpha d^k) = o(\alpha). \tag{2.19}$$

If $v^k_{\max} \neq 0$, since it is independent on $\alpha$, there always exists a $\bar{t}_k \in (0, 1]$ such that (2.8) holds for all $\alpha \in (0, \bar{t}_k]$.

By Lemma 2.4, $(g^k)^T d^k \leq -\gamma\|d^k\|^2$. From this inequality, relation (2.19) and the fact that $v(x^k + \alpha d^k) \geq 0$ for all $\alpha \geq 0$, we can find a small $\tilde{t}_k \in (0, \bar{t}_k]$ such that, for all $\alpha \in (0, \tilde{t}_k]$,

$$\min\{\sigma\alpha(g^k)^T d^k, -\xi_1 v(x^k + \alpha d^k)\} = \sigma\alpha(g^k)^T d^k.$$

Since $f(x^k + \alpha d^k) - f^k - \sigma\alpha(g^k)^T d^k = (1 - \sigma)\alpha(g^k)^T d^k + o(\alpha) \leq 0$ for all sufficiently small $\alpha > 0$, there exists a scalar $t_k \in (0, \tilde{t}_k]$ such that (2.7) and (2.8) hold for all $\alpha \in (0, t_k]$. ☐

The line search procedure (2.7) is motivated by Chin and Fletcher [9] and Fletcher, Leyffer and Toint [14]. The following result partially reflects its role in the algorithm.

**Lemma 2.6** *Suppose that $\{x^k\}$ is the infinite sequence generated by Algorithm 2.2. If (2.7) holds for all $k$, and $\liminf_{k\to\infty} f^k > -\infty$, then*

$$\lim_{k\to\infty} v^k = 0. \tag{2.20}$$

*Proof.* If (2.7) holds for all $k$, then $f^{k+1} \le f^k$ and

$$f^{k+1} - f^k \le -\xi_1 v^{k+1}. \tag{2.21}$$

Since $\{f^k\}$ is a monotonically decreasing sequence and is bounded below, then it converges as $k \to \infty$. Let $k \to \infty$ and take the limit on the two sides of the inequality (2.21), we derive

$$\lim_{k\to\infty} v^{k+1} = 0.$$

Thus, the result (2.20) follows directly. □

**Lemma 2.7** *Suppose that $\{x^k\}$ is the infinite sequence generated by Algorithm 2.2. If the objective function is bounded below, that is, $\liminf_{k\to\infty} f^k > -\infty$, then there exists a constant $\nu_0 > 0$ such that $x^k \in \mathcal{C}$ for all $k \ge 0$, where $\mathcal{C} = \{x : v(x) \le \nu_0\}$ is a level set of $v(x)$.*

*Proof.* If $\liminf_{k\to\infty} f^k > -\infty$ and all iterations are f-type iterations, then by Lemma 2.6 $\lim_{k\to\infty} v^k = 0$. Thus, for any given $\nu > 0$, there is an integer $k_0 > 0$ such that $v^k \le \nu$ for all $k \ge k_0$. Let $\nu_0 = \max\{v^0, v^1, \ldots, v^{k_0}, \nu\}$. Then $v^k \le \nu_0$ for all $k \ge 0$.

If all iterations are h-type iterations, by Algorithm 2.2, $v^1_{\max} = v^0$, $v^k_{\max}$ remains constant for all $k \ge 1$ and (2.9) holds for all $k \ge 0$. If we set $\nu_0 = v^0$, then $v^k \le \nu_0$ for every $k \ge 0$.

Now we consider the last possible case where a set of f-type iterations and a set of h-type iterations appear alternately. If the algorithm starts with an h-type iteration, then we can still set $\nu_0 = v^0$ since by (2.8) and (2.9) we have that $v^k \le v^0$ for all $k \ge 0$. Otherwise, suppose that the first $k_0$ iterations are f-type iterations, and after that, that there are some h-type iterations. Then we can select $\nu_0 = \max\{v^0, v^1, \ldots, v^{k_0}\}$ and there holds $v^k \le \nu_0$ for all $k \ge 0$. Thus, $x^k \in \mathcal{C}$ for all $k \ge 0$. □

## 3. Global convergence

We assume that the algorithm does not terminate in a finite number of iterations and an infinite sequence $\{x^k\}$ is generated. Thus, $\|d^k\| \ne 0$ for all $k \ge 0$.

The following assumptions are based on our discussions in the previous section.

**Assumption 3.1**

*(A) $f$ and $h_j, j \in \mathcal{E}$, are twice continuously differentiable on $\Re^n$;*

*(B) $\liminf_{k\to\infty} f^k > -\infty$, and the level set $\mathcal{C} = \{x \in \Re^n : v(x) \le \nu_0\}$ is bounded, where $\nu_0$ is the value of $v(x)$ at the first h-type iteration, if any, and can be any given positive value if all iterations are f-type iterations;*

*(C) $\|B_k\| \le \beta$ for some scalar $\beta > 0$ and $d^T B_k d \ge \gamma \|d\|^2$, $\forall d \ne 0$ such that $A_k^T d = 0$ for some constant $\gamma > 0$, that is, $B_k$ is bounded above and is positive definite in the null space of $A_k^T$ for every $k$.*

*(D) $d_p^k$ satisfies* **Conditions (a)–(b)** *for all $k \ge 0$.*

Assumption 3.1 (B) can be further simplified if we impose a condition on the initial point $x^0$ that $v^0$ is large enough so that the algorithm starts by an h-type iteration at $x^0$. In this case we can choose $\nu_0 = v^0$ by the proof of Lemma 2.7. But the imposition will affect the freedom in the choice of the initial point and the use of the standard initial point when implementing the algorithm.

It should be noticed that in Assumption 3.1 the usual assumption on the boundedness of the iterative sequence $\{x^k\}$ (for example see [1, 5, 8, 14, 19] et al.) is replaced by Assumption 3.1 (B). To develop algorithms and convergence analysis without requiring the boundedness of the iterative sequence $\{x^k\}$ directly has become an active topic of research for nonlinear programming, for very recent references, we can see [17, 18, 25, 32]. As far as we know, Assumption 3.1 (C) is the mildest assumption on approximate Hessian $B_k$ in line search methods for nonlinear programming in literature. Actually, there are still many newly presented line search algorithms for nonlinear programming requiring $B_k$ to be uniformly positive definite on $\Re^n$.

If Assumption 3.1 (B) holds, then it follows from Lemma 2.7 and its proof that $x^k \in \mathcal{C}$ for all sufficiently large $k$, implying that $\{x^k\}$ is bounded. Thereby, $\|A_k A_k^T\| \leq \kappa$ for some constant $\kappa \geq 1$. Moreover, there is a constant $M > 0$ such that $\|g^k\| \leq M, \|A_k\| \leq M, \|\nabla^2 f^k\| \leq M, \|\nabla^2 h_j^k\| \leq M$ $(\forall j \in \mathcal{E})$.

**Lemma 3.2** *Suppose that Assumption 3.1 holds. Then $\{d^k\}$ and $\{A_k\lambda^k\}$ are bounded, where $(d^k, \lambda^k)$ is a KKT pair of problem (2.1)–(2.2).*

*Proof.* If Assumption 3.1 holds, then **Condition (a)** implies that $\|d_p^k\|$ is bounded. Since

$$(g^k)^T d^k + (1/2)(d^k)^T B_k d^k \leq (g^k)^T d_p^k + (1/2)(d_p^k)^T B_k d_p^k,$$

one has

$$(1/2)(d^k - d_p^k)^T B_k (d^k - d_p^k) \leq (g^k)^T (d_p^k - d^k) + (d_p^k)^T B_k (d_p^k - d^k)$$

with $A_k^T(d^k - d_p^k) = 0$. Thus, by Assumption 3.1 (C),

$$(\gamma/2)\|d^k - d_p^k\|^2 \leq (g^k)^T (d_p^k - d^k) + (d_p^k)^T B_k (d_p^k - d^k). \tag{3.1}$$

If $\{d^k\}$ is unbounded, there exists a subsequence $\{d^k : k \in \mathcal{K}\}$ satisfying $\|d^k\| \to \infty$ as $k \to \infty$ and $k \in \mathcal{K}$, which implies that $\|d^k - d_p^k\| \to \infty$ as $k \to \infty$ and $k \in \mathcal{K}$. Divide by $\|d^k - d_p^k\|^2$ on the two sides of (3.1) and take the limit as $k \to \infty$ and $k \in \mathcal{K}$, we have $\gamma/2 \leq 0$, which contradicts the assumption $\gamma > 0$. The contradiction shows that the claim on the boundedness of $\{d^k\}$ is correct.

The claim that $\{A_k\lambda^k\}$ is bounded follows from (2.16) and the boundednesses of $g^k, B_k$ and $d^k$ directly. □

The following simple result prepares Lemma 3.4.

**Lemma 3.3** *Let $\psi(\alpha) = c_1 + c_2\alpha^2 - c_3\alpha$, where $c_1 \geq 0, c_2 > 0, c_3 > 0$. Then $c_1 + c_2\alpha^2 \leq c_3\alpha$ (that is, $\psi(\alpha) \leq 0$) provided that $c_1 \leq c_3^2/(8c_2)$ and $\alpha \in [2(c_1/c_3), (1/2)(c_3/c_2)]$. In particular, if $c_1 = 0$, then $\psi(\alpha) \leq 0$ as $\alpha \in [0, (1/2)c_3/c_2]$.*

*Proof.* By reformulating, we have $\psi(\alpha) = c_2(\alpha - c_3/(2c_2))^2 + c_1 - c_3^2/(4c_2)$. If $c_1 \le c_3^2/(8c_2)$, then $c_1 - c_3^2/(4c_2) \le -c_3^2/(8c_2)$. Thus, $\psi(\alpha) \le 0$ if $(\alpha - c_3/(2c_2))^2 \le c_3^2/(4c_2^2) - c_1/c_2$. The above condition is fulfilled if

$$c_3/(2c_2) - \sqrt{c_3^2/(4c_2^2) - c_1/c_2} \le \alpha \le c_3/(2c_2) + \sqrt{c_3^2/(4c_2^2) - c_1/c_2}. \tag{3.2}$$

Since

$$\begin{aligned} & c_3/(2c_2) - \sqrt{c_3^2/(4c_2^2) - c_1/c_2} \\ = & \frac{c_1/c_2}{c_3/(2c_2) + \sqrt{c_3^2/(4c_2^2) - c_1/c_2}} \\ \le & \frac{c_1/c_2}{c_3/(2c_2)} \le 2c_1/c_3 \end{aligned}$$

and $c_3/(2c_2) + \sqrt{c_3^2/(4c_2^2) - c_1/c_2} \ge (1/2)(c_3/c_2)$, by (3.2), we complete the proof. □

**Lemma 3.4** *Suppose that Assumption 3.1 holds. If for all $k \ge 0$*

$$\|A_k h^k\| \ge \eta \|h^k\| \tag{3.3}$$

*on some constant $\eta \in (0, 1)$, and $\xi_1 \le \min\{\sigma/(2(1-\sigma)), \min(0.05, (1-\sigma)/16)\sigma\kappa_2\eta^2\gamma/M\}$ and $\xi_2 \le (\sigma/(1-\sigma))M$ with positive constants $M, \gamma, \kappa_2, \sigma$ (where $\gamma$, $\kappa_2$ are constants defined respectively in Assumption 3.1 (C) and* **Condition (b)**, *and $\sigma$ is given in Algorithm 2.2), then there exists a scalar $\tilde{t} \in (0, 1]$ independent of $k$ such that $1 \ge \alpha_k \ge \tilde{t}$ for all $k$. Thus, we have that either both (2.7) and (2.8) are satisfied or (2.9) holds with $\alpha_k$ bounded away from zero.*

*Proof.* Suppose that there is some constant $\epsilon > 0$ such that $\|h^k\| \ge \epsilon \|d^k\|^2 > 0$. By (3.3) and **Condition (b)**,

$$\phi(x^k; d^k) = \|h^k + A_k^T d^k\| - \|h^k\| \le -\kappa_2\eta^2\|h^k\| \le -\kappa_2\eta^2\epsilon\|d^k\|^2. \tag{3.4}$$

Thus, $\alpha\sigma\phi(x^k; d^k) \le -\alpha^2\xi_2\|d^k\|^2$ as $0 < \alpha \le \sigma\kappa_2\eta^2\epsilon/\xi_2$. It follows from (2.13), Assumption 3.1 (B) and (3.4) that

$$\begin{aligned} & v(x^k + \alpha d^k) - v^k - \alpha\sigma\phi(x^k; d^k) \\ & \le \alpha\phi(x^k; d^k) + (1/2)\alpha^2 \sum_{j=1}^{m} \|\nabla^2 h_j(x^k + \theta d^k)\| \|d^k\|^2 - \alpha\sigma\phi(x^k; d^k) \\ & \le (1-\sigma)\alpha\phi(x^k; d^k) + M\alpha^2\|d^k\|^2 \\ & \le -(1-\sigma)\alpha\kappa_2\eta^2\|h^k\| + M\alpha^2\|d^k\|^2 \\ & \le (-(1-\sigma)\epsilon\kappa_2\eta^2 + \alpha M)\alpha\|d^k\|^2, \end{aligned} \tag{3.5}$$

where $M > 0$ is a constant such that $(1/2)\sum_{j=1}^{m} \|\nabla^2 h_j(x^k + \theta d^k)\| \le M$, $\forall \theta \in [0, 1]$. Let $t_1 = \min\{(1-\sigma)\epsilon\kappa_2\eta^2/M, \sigma\kappa_2\eta^2\epsilon/\xi_2\}$, then (2.9) holds for all $\alpha \in [0, t_1]$. Moreover, $t_1 = (1-\sigma)\epsilon\kappa_2\eta^2/M$ when $\xi_2 \le (\sigma/(1-\sigma))M$.

We now consider that $\|h^k\| < \epsilon\|d^k\|^2$ for some given $\epsilon > 0$. Assumption 3.1 (C) suggests that $(d^k - d_p^k)^T B_k(d^k - d_p^k) \geq \gamma\|d^k - d_p^k\|^2$. **Condition (a)** and the inequality $\|A_k\| \leq M$ imply that $\|d_p^k\| \leq \kappa_1\|A_k h^k\| \leq \delta\|h^k\|$ with $\delta = \kappa_1 M$. Thus, based on the boundednesses of $\{\|B_k\|\}$ and $\{\|d^k\|\}$, we have

$$\begin{aligned}(d^k)^T B_k d^k &\geq \gamma\|d^k - d_p^k\|^2 + 2(d^k)^T B_k d_p^k - (d_p^k)^T B_k d_p^k \\ &\geq \gamma\|d^k\|^2 - \gamma'\|h^k\|, \end{aligned} \tag{3.6}$$

where $\gamma' > 0$ is a constant. By (2.16) and the equality $A_k^T d^k = A_k^T d_p^k$,

$$(g^k)^T d^k + (d^k)^T B_k d^k = (d_p^k)^T A_k \lambda^k. \tag{3.7}$$

Hence, it follows from (3.7), (3.6), and Lemma 3.2 that

$$\begin{aligned}(g^k)^T d^k &= (d_p^k)^T A_k \lambda^k - (d^k)^T B_k d^k \\ &\leq \|d_p^k\|\|A_k\lambda^k\| + \gamma'\|h^k\| - \gamma\|d^k\|^2 \\ &\leq \gamma''\|h^k\| - \gamma\|d^k\|^2 \\ &\leq (\gamma''\epsilon - \gamma)\|d^k\|^2 \end{aligned} \tag{3.8}$$

for some constant $\gamma'' > 0$. This gives rise to $(g^k)^T d^k \leq -(\gamma/2)\|d^k\|^2$ provided $\epsilon \leq \gamma/(2\gamma'')$. Due to

$$\begin{aligned}& f(x^k + \alpha d^k) - f^k - \alpha\sigma(g^k)^T d^k \\ & \leq (1-\sigma)\alpha(g^k)^T d^k + (1/2)\alpha^2\|\nabla^2 f(x^k + \theta d^k)\|\|d^k\|^2 \\ & \leq (-(1-\sigma)\gamma/2 + M\alpha)\alpha\|d^k\|^2, \end{aligned}$$

where $\theta \in (0, \alpha]$ and $M$ is a constant such that $(1/2)\|\nabla^2 f(x^k + \theta d^k)\| \leq M \ \forall\theta \in [0, 1]$, if $t_2 = (1-\sigma)\gamma/(2M)$, then

$$f(x^k + \alpha d^k) - f^k \leq \alpha\sigma(g^k)^T d^k \tag{3.9}$$

for all $\alpha \in [0, t_2]$.

From the Taylor's Theorem, there is a $\theta \in (0, \alpha]$ such that

$$\begin{aligned} v(x^k + \alpha d^k) &\leq \|h^k + \alpha A_k^T d^k\| + (1/2)\alpha^2 \sum_{j=1}^{m} \|\nabla^2 h_j(x^k + \theta d^k)\|\|d^k\|^2 \\ &\leq \|h^k\| + \alpha(\|h^k + A_k^T d^k\| - \|h^k\|) + M(\alpha^2\|d^k\|^2) \\ &\leq \epsilon\|d^k\|^2 + M\alpha^2\|d^k\|^2 \end{aligned} \tag{3.10}$$

where $M$ is a constant large enough such that both (3.5) and (3.9) hold. By Lemma 3.3, if $\alpha \in [4\xi_1\epsilon/(\sigma\gamma),\ \sigma\gamma/(4\xi_1 M)]$ and $\epsilon \leq \sigma^2\gamma^2/(32M\xi_1^2)$, then

$$\xi_1 v(x^k + \alpha d^k) \leq \sigma\alpha(\gamma/2)\|d^k\|^2. \tag{3.11}$$

Thus, it is proved that, if

$$\epsilon \leq \min\{\sigma^2\gamma^2/(32M\xi_1^2), \gamma/(2\gamma^{''})\}, \tag{3.12}$$

then $-\xi_1 v(x^k + \alpha d^k) \geq -\sigma\alpha(\gamma/2)\|d^k\|^2 \geq \sigma\alpha(g^k)^T d^k$ provided $\alpha \in [4\xi_1\epsilon/(\sigma\gamma),\ \sigma\gamma/(4\xi_1 M)]$.

If $\xi_1 \leq \sigma/(2(1-\sigma))$, then $t_2 \leq \sigma\gamma/(4\xi_1 M)$. As long as $\epsilon \leq \sigma(1-\sigma)\gamma^2/(32M\xi_1)$, we have $t_2 \geq 16\xi_1\epsilon/(\sigma\gamma)$, which eventually means that (2.7) holds with $\alpha \in [4\xi_1\epsilon/(\sigma\gamma), t_2]$ when

$$\epsilon \leq \min\{\sigma(1-\sigma)\gamma^2/(32M\xi_1), \sigma^2\gamma^2/(32M\xi_1^2), \gamma/(2\gamma^{''})\}. \tag{3.13}$$

We have $t_1 \geq 16\xi_1\epsilon/(\sigma\gamma)$ provided $\xi_1 \leq \sigma(1-\sigma)\kappa_2\eta^2\gamma/(16M)$. Let $\hat{t} = \min\{t_1, t_2, 1\}$. Finally, at least one of (2.7) and (2.9) holds with $\alpha \in [4\xi_1\epsilon/(\sigma\gamma), \hat{t}]$ for every $k \geq 0$.

In the remnant of the proof, we consider (2.8) with $\|h^k\| \leq \epsilon\|d^k\|^2$. According to the setting of $v_{\max}^k$ in the algorithm and the above discussions, without loss of generality, we assume that $v_{\max}^k \geq \epsilon\|d^k\|^2$ for some given $\epsilon > 0$ satisfying (3.13). For the trivial case $\|h^k\| = 0$, **Condition (a)** implies that $\|d_p^k\| = 0$, which gives rise to $A_k^T d^k = 0$. Hence, by the second inequality of (3.10), together with (2.2) and (2.10), as $\alpha \in [0, \sqrt{\epsilon/(2M)}]$, we have

$$v(x^k + \alpha d^k) \leq M(\alpha^2\|d^k\|^2) \leq (1/2)\epsilon\|d^k\|^2 \leq ((r_k+1)/2)v_{\max}^k.$$

In what follows, we assume that $\|h^k\| \neq 0$. If $x^k$ is generated by an f-type iteration, then $\|h^k\| \leq \max\{(r_k+1)/2, 0.95\}v_{\max}^k$. Thus, it follows from the second inequality of (3.10) and the first inequality of (3.4) that

$$\begin{aligned} v(x^k + \alpha d^k) &\leq (1 - \alpha\kappa_2\eta^2)\|h^k\| + \alpha^2 M\|d^k\|^2 \\ &\leq (1 - \alpha\kappa_2\eta^2)\max\{(r_k+1)/2, 0.95\}v_{\max}^k + \alpha^2(M/\epsilon)v_{\max}^k \\ &\leq \max\{(r_k+1)/2, 0.95\}v_{\max}^k - 0.95\alpha\kappa_2\eta^2 v_{\max}^k + \alpha^2(M/\epsilon)v_{\max}^k. \end{aligned}$$

Therefore, (2.8) holds provided that we select $\alpha \in [0, 0.95\kappa_2\eta^2\epsilon/M]$.

If $x^k$ is obtained by an h-type iteration, then $\|h^k\| = r_k\|h^{k-1}\| \leq r_k v_{\max}^k$. As $\alpha \in [0, 0.9\kappa_2\eta^2\epsilon/M]$, (2.8) holds since, if $r_k \geq 0.9$, $\max\{(r_k+1)/2, 0.95\} = (r_k+1)/2$ and

$$\begin{aligned} v(x^k + \alpha d^k) &\leq (1 - \alpha\kappa_2\eta^2)\|h^k\| + \alpha^2 M\|d^k\|^2 \\ &\leq ((1 - \alpha\kappa_2\eta^2)r_k + \alpha^2 M/\epsilon)v_{\max}^k \\ &\leq (r_k - 0.9\alpha\kappa_2\eta^2 + \alpha^2 M/\epsilon)v_{\max}^k \\ &\leq r_k v_{\max}^k \\ &< ((r_k+1)/2)v_{\max}^k \quad (\text{by} \quad (2.10)), \end{aligned}$$

otherwise, $r_k < 0.9$, $\max\{(r_k+1)/2, 0.95\} = 0.95$ and

$$\begin{aligned} v(x^k + \alpha d^k) &\leq ((1 - \alpha\kappa_2\eta^2)r_k + \alpha^2 M/\epsilon)v_{\max}^k \\ &< (0.9 - 0.9\alpha\kappa_2\eta^2 + \alpha^2 M/\epsilon)v_{\max}^k \\ &< 0.95v_{\max}^k. \end{aligned}$$

If $\xi_1 \le 0.05\sigma\kappa_2\eta^2\gamma/M$, then $4\xi_1\epsilon/(\sigma\gamma) \le 0.2\kappa_2\eta^2\epsilon/M$. There holds $4\xi_1\epsilon/(\sigma\gamma) \le (1/2)\sqrt{\epsilon/(2M)}$ provided $\epsilon \le \sigma^2\gamma^2/(128M\xi_1^2)$. By summarizing the above discussions, if

$$\xi_1 \le \min\{\sigma/(2(1-\sigma)), \min(0.05, (1-\sigma)/16)\sigma\kappa_2\eta^2\gamma/M\}, \quad \xi_2 \le (\sigma/(1-\sigma))M,$$

we have that either both (2.7) and (2.8) are satisfied or (2.9) holds with $\alpha \in [4\xi_1\epsilon/(\sigma\gamma), \hat{t}_0]$, where $\hat{t}_0 = \min\{\hat{t}, 0.9\kappa_2\eta^2\epsilon/M, \sqrt{\epsilon/(2M)}\}$.

At last, the result follows from selecting $\tilde{t} \in (0, 4\xi_1\epsilon/(\sigma\gamma)]$ with

$$\epsilon \le \min\{\sigma(1-\sigma)\gamma^2/(32M\xi_1), \sigma^2\gamma^2/(128M\xi_1^2), \gamma/(2\gamma^{''})\}.$$

□

**Lemma 3.5** *Suppose that the conditions in Lemma 3.4 hold. If (2.9) holds for all sufficiently large k, then*

$$\lim_{k\to\infty} v^k = 0 \quad \textit{and} \quad \lim_{k\to\infty} \|d^k\| = 0. \tag{3.14}$$

*Proof.* By Lemma 3.4, if (2.9) holds for iteration $k$, then

$$v^{k+1} - v^k \le \sigma\alpha_k\phi(x^k; d^k) \le -\sigma\alpha_k\kappa_2\eta^2\|h^k\| \le -\eta^{'}\|h^k\|,$$

where $\eta^{'} \in (0,1)$ is a constant. Therefore, by (2.6), $v^{k+1} \le (1-\eta^{'})v^k$, which implies that $\lim_{k\to\infty} v^k = 0$.

Again by (2.9),

$$v^{k+1} - v^k \le -\xi_2\alpha_k^2\|d^k\|^2 \le -\xi_2\tilde{t}^2\|d^k\|^2 \le 0. \tag{3.15}$$

By taking the limit on the two sides of (3.15), we have $\lim_{k\to\infty}\|d^k\| = 0$. □

**Lemma 3.6** *Suppose that the conditions in Lemma 3.4 hold. If (2.7) is satisfied for all sufficiently large k, then*

$$\lim_{k\to\infty} v^k = 0 \quad \textit{and} \quad \lim_{k\to\infty} \|d^k\| = 0.$$

*Proof.* If for all sufficiently large $k$, (2.7) holds, then $\{f^k\}$ is monotonically decreasing after a finite number of iterations. Since

$$f^{k+1} - f^k \le -\xi_1 v^{k+1},$$

by taking the limit on the two sides of the above inequality, we have $\lim_{k\to\infty} v^k = 0$.

Assume now that $\|d^k\| \not\to 0$ for $k \in \mathcal{K}$, where $\mathcal{K}$ is an infinite subset of the index set. Since $\lim_{k\to\infty} v^k = 0$, it follows from (3.8) that $(g^k)^T d^k \le -\gamma_0\|d^k\|^2$ for some constant $\gamma_0 > 0$ and $k \in \mathcal{K}$. Thus, for sufficiently large $k \in \mathcal{K}$,

$$f^{k+1} - f^k \le \sigma\alpha_k(g^k)^T d^k \le -\sigma\tilde{t}\gamma_0\|d^k\|^2. \tag{3.16}$$

Together with the fact that $\{f^k\}$ is monotonically decreasing and bounded below, (3.16) implies that $\lim_{k\to\infty, k\in\mathcal{K}}\|d^k\| = 0$, which is a contradiction. Thus, the result is obtained. □

**Lemma 3.7** *Suppose that the conditions in Lemma 3.4 hold. Then*

$$\lim_{k\to\infty} v^k = 0 \quad \textit{and} \quad \liminf_{k\to\infty} \|\nabla_x L(x^k,\lambda^k)\| = 0. \tag{3.17}$$

*Proof.* If $\lim_{k\to\infty}\|d^k\| = 0$, by (2.16), we have

$$\lim_{k\to\infty} \|\nabla_x L(x^k,\lambda^k)\| = \lim_{k\to\infty} \|B_k d^k\| = 0.$$

If all except for a finite number of iterations are h-type iterations or f-type iterations, then, by Lemma 3.5 and Lemma 3.6, (3.17) is obtained.

We therefore need only to consider the case where h-type iterations and f-type iterations appear alternately and constantly. Without loss of generality, assume that the iterations from the $(k_{(2\ell-1)}+1)$th iteration to the $k_{2\ell}$th iteration and from the $(k_{(2\ell+1)}+1)$th iteration to the $k_{(2\ell+2)}$th iteration are $h$-iterations, and that those from the $(k_{2\ell}+1)$th iteration to the $k_{(2\ell+1)}$th iteration are $f$-iterations, where $\ell = 0, 1, 2, \ldots$.

It will be firstly proved that

$$\lim_{k\to\infty} v^k = 0. \tag{3.18}$$

Lemma 3.4 shows that there is a constant $\tilde{t} > 0$ such that $\alpha_k \geq \tilde{t}$ for all $k \geq 0$. Thus, by (2.9), **Condition (b)** and (3.3), for any h-iteration at $x^j$,

$$v^{j+1} - v^j \leq \sigma\alpha_j\phi(x^j; d^j) \leq -\sigma\tilde{t}\kappa_2\eta^2 v^j,$$

which gives rise to that $r_k \leq \hat{r} = 1 - \sigma\tilde{t}\kappa_2\eta^2 < 1$ for all $k \geq 0$. Hence,

$$\max\{(r_k+1)/2, 0.95\} \leq \max\{(\hat{r}+1)/2, 0.95\} < 1 \quad \text{for all } k \geq 0.$$

Let $\tilde{r} = \max\{(\hat{r}+1)/2, 0.95\}$. By the algorithm, for $\ell = 0, 1, 2, \ldots$,

$$v^{k_{(2\ell-1)}} > v^{(k_{(2\ell-1)}+1)} > \ldots > v^{k_{2\ell}}, \tag{3.19}$$

$$v^{k_{(2\ell+1)}} > v^{(k_{(2\ell+1)}+1)} > \ldots > v^{k_{(2\ell+2)}}, \tag{3.20}$$

and

$$f^{(k_{2\ell}+1)} > f^{(k_{2\ell}+2)} > \ldots > f^{k_{(2\ell+1)}},$$

$$v^{(k_{2\ell}+1)} \leq \tilde{r}v^{k_{(2\ell-1)}}, v^{(k_{2\ell}+2)} \leq \tilde{r}v^{k_{(2\ell-1)}}, \ldots, v^{k_{(2\ell+1)}} \leq \tilde{r}v^{k_{(2\ell-1)}}. \tag{3.21}$$

That is, the subsequence $\{\ldots, v^{k_{(2\ell-1)}}, v^{k_{(2\ell+1)}}, \ldots\}$ is a monotonically decreasing sequence with the ratio $v^{k_{(2\ell+1)}}/v^{k_{(2\ell-1)}} \leq \tilde{r} < 1$ for all $\ell \geq 0$. At last, we have

$$\lim_{\ell\to\infty} v^{k_{(2\ell+1)}} = 0.$$

Combining with (3.19), (3.20), (3.21) and the nonnegativity of $v^k$, (3.18) follows immediately.

In what follows, we prove that there is a subsequence $\{d^k : k \in \mathcal{K}\}$ such that

$$\lim_{k\in\mathcal{K},k\to\infty} \|d^k\| = 0. \tag{3.22}$$

If this is not the case, then $\|d^k\| \geq \epsilon$ for all sufficiently large integer $k$ and for some small positive constant $\epsilon$. Thus, by Lemma 3.4,

$$-\xi_2\alpha_k^2\|d^k\|^2 \leq -\xi_2\tilde{t}^2\epsilon^2. \tag{3.23}$$

On the other hand, since $0 < \alpha_k \leq 1$, $\sigma\alpha_k\phi(x^k; d^k) \geq -\sigma\alpha_k\|h^k\| \geq -\sigma v^k$. Therefore, provided $v^k \leq \xi_2\tilde{t}^2\epsilon^2/\sigma$, then

$$\sigma\alpha_k\phi(x^k; d^k) \geq -\xi_2\alpha_k^2\|d^k\|^2. \tag{3.24}$$

If (2.9) holds, it follows from (3.24) and (3.23) that $v^{k+1} - v^k \leq -\xi_2\alpha_k^2\|d^k\|^2 \leq -\xi_2\tilde{t}^2\epsilon^2$, which by (3.18) is a contradiction. This contradiction shows that (2.7) and (2.8) hold for all sufficiently large $k$. Hence, by Lemma 3.6, $\lim_{k\to\infty} \|d^k\| = 0$, which contradicts that $\|d^k\| \geq \epsilon$ for all sufficiently large integer $k$. The contradiction shows that (3.22) holds on some set $\mathcal{K}$.

From (2.16) and (3.22) we can deduce that

$$\lim_{k\in\mathcal{K},k\to\infty} \|\nabla_x L(x^k, \lambda^k)\| = \lim_{k\in\mathcal{K},k\to\infty} \|B_k d^k\| = 0.$$

Thus, the results in (3.17) have been proved. □

The above lemma has proved that under suitable conditions the algorithm will terminate at an approximate KKT point of the original problem. The following theorem shows that in a more general situation without requiring (3.3) the presented algorithm has strong global convergence properties.

**Theorem 3.8** *Suppose that Assumption 3.1 holds, and that $\xi_1$ and $\xi_2$ are taken as such in Lemma 3.4. If the tolerance $\epsilon > 0$ in Algorithm 2.2 is small, then the algorithm terminates in a finite number of iterations at either an approximate KKT point, or an approximate infeasible stationary point or an approximate feasible point at which the LICQ (or MFCQ) does not hold.*

*Proof.* Assume that the algorithm does not terminate in a finite number of iterations. We need to consider two cases:

**Case (i).** $\|A_k h^k\| \geq \eta\|h^k\|$ for all sufficiently large $k$ and for some constant $\eta > 0$. In this case, Lemma 3.7 shows that the algorithm will terminate since the condition

$$\|\nabla_x L(x^k, \lambda^k)\| \leq \epsilon \quad \text{and} \quad \|h^k\| \leq \epsilon$$

will be satisfied at some iterate $x^k$. For small $\epsilon$, the termination point is an approximate KKT point of problem (1.1)–(1.2).

**Case (ii).** There is an infinite index set $\mathcal{K}$ such that $\lim_{k\in\mathcal{K},k\to\infty} \|A_k h^k\|/\|h^k\| = 0$. In this case, the algorithm will terminate at a point $x^k$ where

$$\|A_k h^k\| \le \epsilon \min(\|h^k\|, 1).$$

For small $\epsilon$, we need further to investigate two subcases:
**Subcase (ii-a).** $\|h^k\|$ is small enough. In this subcase, $\|h^k\| < 1$ and $\|A_k h^k\|/\|h^k\| \le \epsilon$. Let $w^k = h^k/\|h^k\|$. Then $\|A_k w^k\| \le \epsilon$. That is, there exists a unitary vector $w^k \in \Re^m$ such that $\|A_k w^k\|$ is small enough. Thus, $x^k$ is an approximate feasible point at which the LICQ (or MFCQ) does not hold.
**Subcase (ii-b).** $\|h^k\|$ is bounded away from zero, for instance, $\|h^k\| > 1$. In this subcase, $\|A_k h^k\| = O(\epsilon)$, $x^k$ can be thought as an approximate stationary point of minimizing the constraint violations

$$\min \quad \|h(x)\|^2.$$

Thus, $x^k$ is an approximate infeasible stationary point of problem (1.1)–(1.2). □

## 4. Local analysis

In order to obtain locally rapid convergence, we need to overcome the so-called Maratos effect, a phenomenon arising in many methods for nonlinear programming where the full superlinear step is rejected near the solution. Thus, we introduce a second-order correction technique (see [11, 22]) in Algorithm 2.2 and derive the following algorithm.

**Algorithm 4.1** *(The algorithm with second-order correction)*

**Given** *initial point* $x^0 \in \Re^n$, *constant* $\sigma \in (0, 1/2)$, *small positive constants* $\xi_1, \xi_2$, $v_{soc}$ *and the tolerance* $\epsilon > 0$. *Compute* $g^0$, $h^0$, $A_0$ *and* $B_0$. *Set* $v_{\max}^0 = 0, r_0 = 0.9$. *Let* $k := 0$;

**While** $\max(\|\nabla_x L(x^k, \lambda^k)\|, \|h^k\|) > \epsilon$ *and* $\|A_k h^k\| > \epsilon \min(\|h^k\|, 1)$;

**Calculate** $d_p^k$ *approximately minimizing* $\|h^k + A_k^T d\|$ *on* $d$ *satisfying* **Conditions (a)–(b)**.

**Solve** *the subproblem (2.1)–(2.2). Let* $d^k$ *be the solution.*

**If** *both*

$$f(x^k + d^k) - f^k \le \min\{\sigma (g^k)^T d^k, -\xi_1 v(x^k + d^k)\} \tag{4.1}$$

*and*

$$v(x^k + d^k) \le \max\{(r_k + 1)/2, 0.95\} v_{\max}^k \quad \text{if} \quad v_{\max}^k \ne 0 \tag{4.2}$$

*hold, or inequality*

$$v(x^k + d^k) - v^k \le \min\{\sigma \phi(x^k; d^k), -\xi_2 \|d^k\|^2\} \tag{4.3}$$

*is satisfied, set* $x^{k+1} = x^k + d^k$;

**else if** $v^k \le v_{soc}$, *calculate* $\tilde{d}_p^k$ *approximately minimizing* $\|h(x^k+d^k)+A_k^T d\|$ *on* $d$ *satisfying* **Conditions (a)–(b)**. *Solve the correction subproblem*

$$\min \;\; (g^k)^T(d^k+d)+(1/2)(d^k+d)^T B_k(d^k+d) \tag{4.4}$$

$$s.t. \;\; A_k^T d = A_k^T \tilde{d}_p^k \tag{4.5}$$

*to obtain* $\tilde{d}^k$, *and check if*

$$f(x^k+d^k+\tilde{d}^k)-f^k \le \min\{\sigma(g^k)^T d^k, -\xi_1 v(x^k+d^k+\tilde{d}^k)\}, \tag{4.6}$$

$$v(x^k+d^k+\tilde{d}^k) \le \max\{(r_k+1)/2, 0.95\} v_{\max}^k \quad \text{if} \quad v_{\max}^k \ne 0 \tag{4.7}$$

*If both (4.6) and (4.7) are satisfied, set* $x^{k+1}=x^k+d^k+\tilde{d}^k$;
**else** *select* $\alpha_k \in (0,1)$ *as large as possible such that either both inequalities*

$$f(x^k+\alpha_k d^k)-f^k \le \min\{\sigma\alpha_k(g^k)^T d^k, -\xi_1 v(x^k+\alpha_k d^k)\} \tag{4.8}$$

*and*

$$v(x^k+\alpha_k d^k) \le \max\{(r_k+1)/2, 0.95\} v_{\max}^k \quad \text{if} \quad v_{\max}^k \ne 0 \tag{4.9}$$

*hold, or inequality*

$$v(x^k+\alpha_k d^k)-v^k \le \min\{\sigma\alpha_k\phi(x^k;d^k), -\xi_2\alpha_k^2\|d^k\|^2\} \tag{4.10}$$

*is satisfied. Set* $x^{k+1}=x^k+\alpha_k d^k$.
**If** *either (4.3) or (4.10) holds at* $x^{k+1}$ *but not* $x^k$, *set* $v_{\max}^{k+1}=v^k$, *else* $v_{\max}^{k+1}=v_{\max}^k$;
**Compute** $g^{k+1}$, $h^{k+1}$, $v^{k+1}$, $A_{k+1}$ *and* $B_{k+1}$. *If (4.3) or (4.10) holds, calculate* $r_{k+1}=v^{k+1}/v^k$; *otherwise,* $r_{k+1}=r_k$. *Let* $k:=k+1$;

**end (while)**

We use the second-order correction technique only when $v^k$ is small enough, where $v_{soc}$ is introduced for practical implementations so that we only compute the second-order correction step as $x^k$ is close enough to the solution $x^*$. We prove that (4.2) will be satisfied as $k\to\infty$. As a result, the second-order correction procedure will be started if $x^k+d^k$ is not accepted for sufficiently large $k$.

In order to study the local convergence properties of our algorithm, we assume the tolerance $\epsilon=0$ and need some additional assumptions.

**Assumption 4.2**

*(1)* $x^k \to x^*$ *as* $k\to\infty$, *where* $x^*$ *is a KKT point of problem (1.1)–(1.2),* $\lambda^* \in \Re^m$ *is an associated Lagrangian multiplier vector;*

*(2)* $f(x)$ *and* $h(x)$ *are twice differentiable, and their second derivatives are Lipschitz continuous at* $x^*$;

*(3)* $\mathcal{J}_k = \mathcal{J}^*$*, where* $\mathcal{J}^* \subseteq \mathcal{E}$ *and* $\mathcal{J}_k \subseteq \mathcal{E}$ *are index sets consisting of the largest number of linearly independent column vectors of* $\{\nabla h_j^* : j \in \mathcal{E}\}$ *and* $\{\nabla h_j^k : j \in \mathcal{E}\}$*, respectively;*

*(4)* $\lambda_j^* = 0 \ \forall j \notin \mathcal{J}^*$ *and* $j \in \mathcal{E}$*, and* $\lambda_j^k = 0 \ \forall j \notin \mathcal{J}_k$ *and* $j \in \mathcal{E}$*;*

*(5)* $d^T \nabla_{xx}^2 L(x^*, \lambda^*) d \geq \gamma \|d\|^2$, $\forall d \in \{d \neq 0 : (\nabla h_j^*)^T d = 0, j \in \mathcal{J}^*\}$*, where* $L(x, \lambda) = f(x) + \lambda^T h(x)$*, and* $\gamma > 0$ *is a constant.*

If $\nabla h_j^*, j \in \mathcal{E}$, are linearly independent (for example see [2, 6, 31]), we can see that Assumption 4.2 (3) and (4) hold. It follows from Assumption 4.2 (1), (3) and (4) that $\lambda^*$ is unique. Moreover, under Assumption 4.2, if $\|(B_k - \nabla_{xx}^2 L(x^*, \lambda^*))d\| = o(\|d\|)$ at every $d \in \Re^n$, then there exists a constant $\tilde{\gamma} > 0$ such that for all sufficiently large $k$, $d^T B_k d \geq \tilde{\gamma} \|d\|^2$, $\forall d \in \{d \neq 0 : (\nabla h_j^k)^T d = 0, j \in \mathcal{J}_k\}$.

The following results show that under suitable assumptions $d^k$ is a superlinearly or quadratically convergent step.

**Theorem 4.3** *Suppose that Assumption 4.2 holds and* $\|h^k + A_k^T d_p^k\| \leq (1 - \eta_k)\|h^k\|$ $(0 < \eta_k \leq 1)$.

*(1) If* $\|(B_k - \nabla_{xx}^2 L(x^*, \lambda^*))d^k\| = o(\|d^k\|)$*, and* $(1 - \eta_k) = o(1)$*, then*

$$\lim_{k \to \infty} \|x^k + d^k - x^*\| / \|x^k - x^*\| = 0. \tag{4.11}$$

*(2) If* $B_k = \nabla_{xx}^2 L(x^k, \lambda^k)$*, and* $(1 - \eta_k) = O(\|h^k\|)$*, then*

$$\|x^k + d^k - x^*\| = O(\|x^k - x^*\|^2).$$

*Proof.* Let $P_* = I - A_{\mathcal{J}^*}(A_{\mathcal{J}^*}^T A_{\mathcal{J}^*})^{-1} A_{\mathcal{J}^*}^T$ and $P_k = I - A_{\mathcal{J}_k}(A_{\mathcal{J}_k}^T A_{\mathcal{J}_k})^{-1} A_{\mathcal{J}_k}^T$, where $I$ is the $n \times n$ unitary matrix. Consider the system

$$\begin{bmatrix} P_* \nabla_{xx}^2 L(x^*, \lambda^*) \\ A_{\mathcal{J}^*}^T \end{bmatrix} d = 0. \tag{4.12}$$

Let $d^* \in \Re^n$ be any of its solutions. If $d^* \neq 0$, then

$$(d^*)^T P_* \nabla_{xx}^2 L(x^*, \lambda^*) d^* = 0, \quad A_{\mathcal{J}^*}^T d^* = 0,$$

so we have $(d^*)^T \nabla_{xx}^2 L(x^*, \lambda^*) d^* = 0$, which contradicts Assumption 4.2 (5). This contradiction shows that the coefficient matrix of the system (4.12) has full column rank. Therefore, by Assumption 4.2, for all sufficiently large $k$, the matrix

$$\begin{bmatrix} P_k \nabla_{xx}^2 L(x^*, \lambda^*) \\ A_{\mathcal{J}_k}^T \end{bmatrix}$$

is of full column rank.

Since $g^k + B_k d^k + A_{\mathcal{J}_k} \lambda_{\mathcal{J}_k}^k = 0$, there holds

$$\begin{aligned} P_k B_k d^k &= -P_k (g^k + A_{\mathcal{J}_k} \lambda_{\mathcal{J}_k}^*) \\ &= -P_k \nabla_{xx}^2 L(x^*, \lambda^*)(x^k - x^*) + O(\|x^k - x^*\|^2). \end{aligned}$$

Thus,

$$P_k(B_k - \nabla^2_{xx}L(x^*,\lambda^*))d^k = -P_k\nabla^2_{xx}L(x^*,\lambda^*)(x^k + d^k - x^*) + O(\|x^k - x^*\|^2). \tag{4.13}$$

Thanks to $h^k_{\mathcal{J}_k} = h^k_{\mathcal{J}_k} - h^*_{\mathcal{J}_k} = A^T_{\mathcal{J}_k}(x^k - x^*) + O(\|x^k - x^*\|^2)$ and $h^*_{\mathcal{J}_k} = 0$, there holds

$$A^T_{\mathcal{J}_k}(x^k + d^k - x^*) = h^k_{\mathcal{J}_k} + A^T_{\mathcal{J}_k}d^k + O(\|x^k - x^*\|^2). \tag{4.14}$$

Putting (4.13) and (4.14) together in a matrix, we can obtain

$$\begin{aligned}&\begin{bmatrix} P_k\nabla^2_{xx}L(x^*,\lambda^*) \\ A^T_{\mathcal{J}_k} \end{bmatrix}(x^k + d^k - x^*) \\ &= \begin{bmatrix} -P_k(B_k - \nabla^2_{xx}L(x^*,\lambda^*))d^k \\ h^k_{\mathcal{J}_k} + A^T_{\mathcal{J}_k}d^k \end{bmatrix} + O(\|x^k - x^*\|^2),\end{aligned} \tag{4.15}$$

where the coefficient matrix has previously been proved to be of full column rank for all sufficiently large $k$.

(1) Note that $(1-\eta_k) = o(1)$, $\|h^k\| = \|h^k - h^*\| = O(\|x^k - x^*\|)$ and

$$\|h^k_{\mathcal{J}_k} + A^T_{\mathcal{J}_k}d^k\| \le \|h^k + A^T_k d^k\| \le (1-\eta_k)\|h^k\|, \tag{4.16}$$

implying that

$$\|h^k_{\mathcal{J}_k} + A^T_{\mathcal{J}_k}d^k\| = o(\|x^k - x^*\|).$$

If $\|(B_k - \nabla^2_{xx}L(x^*,\lambda^*))d^k\| = o(\|d^k\|)$, then

$$\left\| \begin{bmatrix} -P_k(B_k - \nabla^2_{xx}L(x^*,\lambda^*))d^k \\ h^k_{\mathcal{J}_k} + A^T_{\mathcal{J}_k}d^k \end{bmatrix} \right\| = \sqrt{o(\|d^k\|^2) + o(\|x^k - x^*\|^2)}.$$

Hence, by (4.15) and since $x^k \to x^*$ as $k \to \infty$, there holds

$$\lim_{k\to\infty} \|x^k + d^k - x^*\|/\|x^k - x^*\| = \lim_{k\to\infty} o(\|d^k\|)/\|x^k - x^*\|, \tag{4.17}$$

which shows that

$$\lim_{k\to\infty} \|d^k\|/\|x^k - x^*\| = 1. \tag{4.18}$$

Thus, (4.11) follows from (4.17).

(2) If $B_k = \nabla^2_{xx}L(x^k,\lambda^k)$ and $1-\eta_k = O(\|h^k\|)$, then it follows from (4.16) that

$$\|h^k_{\mathcal{J}_k} + A^T_{\mathcal{J}_k}d^k\| = O(\|x^k - x^*\|^2).$$

Thus,

$$\left\| \begin{bmatrix} -P_k(B_k - \nabla^2_{xx}L(x^*,\lambda^*))d^k \\ h^k_{\mathcal{J}_k} + A^T_{\mathcal{J}_k}d^k \end{bmatrix} \right\| = O(\|x^k - x^*\|^2).$$

The result is obtained immediately from (4.15) due to $x^k \to x^*$ as $k \to \infty$. $\square$

According to (4.18), $d^k \to 0$ as $x^k \to x^*$, and $\|d^k\| = O(\|x^k - x^*\|)$. By (4.11), $\|x^k + d^k - x^*\| = o(\|x^k - x^*\|)$.

**Lemma 4.4** *Suppose that Assumption 4.2 holds, and* $\|(B_k - \nabla^2_{xx}L(x^*, \lambda^*))d^k\| = o(\|d^k\|)$, $\|h^k + A_k^T d^k\| \le (1-\eta_k)\|h^k\|$, $(1-\eta_k) = o(1)$*. If (4.3) does not hold for sufficiently large* $k$*, then*

$$v^k = O(\|d^k\|^2).$$

*Proof.* If (4.3) fails to hold, then we have that either $\sigma\phi(x^k; d^k) \le -\xi_2\|d^k\|^2$ and

$$v(x^k + d^k) - v^k > \sigma\phi(x^k; d^k) \tag{4.19}$$

or $\sigma\phi(x^k; d^k) > -\xi_2\|d^k\|^2$ and

$$v(x^k + d^k) - v^k > -\xi_2\|d^k\|^2. \tag{4.20}$$

As (4.19) holds, since $\phi(x^k; d^k) \ge -v^k$ and

$$\begin{aligned} v(x^k + d^k) &= \|h(x^k + d^k)\| \\ &\le \|h^k + A_k^T d^k\| + O(\|d^k\|^2) \\ &\le (1-\eta_k)v^k + O(\|d^k\|^2), \end{aligned} \tag{4.21}$$

we have $(\eta_k - \sigma)v^k = O(\|d^k\|^2)$. Thus, the result follows from $(1-\eta_k) = o(1)$ and $\sigma < 1$. Otherwise, by (4.20) and (4.21), $\eta_k v^k \le \xi_2\|d^k\|^2 + O(\|d^k\|^2)$, which again by the supposition $(1-\eta_k) = o(1)$ implies the result. Hence, the proof is finished. □

Corresponding to Assumption 4.2 (4), we need the following assumption on the second-order correction subproblem:

**Assumption 4.5** $\tilde{\lambda}_j^k = 0$ $\forall j \notin \mathcal{J}_k$ *and* $j \in \mathcal{E}$*, where* $\tilde{\lambda}^k \in \Re^m$ *is the associated Lagrangian multiplier vector of the correction subproblem (4.4)–(4.5).*

**Lemma 4.6** *Under Assumption 4.2 and Assumption 4.5, if* $\|(B_k - \nabla^2_{xx}L(x^*, \lambda^*))d\| = o(\|d\|)$ *at every* $d \in \Re^n$, $\|h^k + A_k^T d^k\| \le (1-\eta_k)\|h^k\|$ *and* $(1-\eta_k) = o(1)$*, then*

$$\|\tilde{d}^k\| = o(\|d^k\|).$$

*Proof.* The first part of the proof of Theorem 4.3 has demonstrated that the matrix

$$\begin{bmatrix} P_k \nabla^2_{xx}L(x^*, \lambda^*) \\ A_{\mathcal{J}_k}^T \end{bmatrix}$$

has full column rank. Thus, there exists a constant $\hat{\delta} > 0$ such that

$$\left\| \begin{bmatrix} P_k \nabla^2_{xx}L(x^*, \lambda^*) \\ A_{\mathcal{J}_k}^T \end{bmatrix} d \right\| \ge \hat{\delta}\|d\| \tag{4.22}$$

for every $d \in \Re^n$.

If $\|(B_k - \nabla^2_{xx} L(x^*, \lambda^*))d\| = o(\|d\|)$ at every $d \in \Re^n$, thanks to $\|A_k^T d\| \geq \|A_{\mathcal{J}_k}^T d\|$ and

$$\|P_k B_k d\| \geq \|P_k \nabla^2_{xx} L(x^*, \lambda^*) d\| - \|P_k (B_k - \nabla^2_{xx} L(x^*, \lambda^*))d\|,$$

by (4.22), there is a positive constant $\check{\delta} < \hat{\delta}$ such that

$$\left\| \begin{bmatrix} P_k B_k \\ A_k^T \end{bmatrix} d \right\| \geq \check{\delta} \|d\| \tag{4.23}$$

for every $d \in \Re^n$ and for all sufficiently large $k$.

Since $\tilde{d}^k$ solves problem (4.4)–(4.5), then

$$g^k + B_k(d^k + \tilde{d}^k) + A_k \tilde{\lambda}^k = 0, \tag{4.24}$$

where $\tilde{\lambda}^k$ is the associated Lagrangian multiplier vector. Let $\tilde{\mathcal{J}}_k = \mathcal{E} \backslash \mathcal{J}_k$. Assumption 4.2 (4) and Assumption 4.5 imply that $A_{\tilde{\mathcal{J}}_k} \lambda^k_{\tilde{\mathcal{J}}_k} = 0$ and $A_{\tilde{\mathcal{J}}_k} \tilde{\lambda}^k_{\tilde{\mathcal{J}}_k} = 0$. Thus, by (2.16) and (4.24),

$$B_k \tilde{d}^k = A_k(\lambda^k - \tilde{\lambda}^k) = A_{\mathcal{J}_k}(\lambda^k_{\mathcal{J}_k} - \tilde{\lambda}^k_{\mathcal{J}_k}),$$

which gives rise to $P_k B_k \tilde{d}^k = 0$. Hence, using (4.23), we have

$$\check{\delta} \|\tilde{d}^k\| \leq \|A_k^T \tilde{d}^k\|. \tag{4.25}$$

The **Condition (a)** suggests that $\|\tilde{d}^k_p\| \leq \kappa_1 \|A_k h(x^k + d^k)\|$, which induces to

$$\begin{aligned} \|A_k^T \tilde{d}^k\| &= \|A_k^T \tilde{d}^k_p\| \\ &\leq \kappa_1 \|A_k^T\| \|A_k\| \|h(x^k + d^k)\| \\ &\leq \kappa_1 \|A_k^T\| \|A_k\| (\|h^k + A_k^T d^k\| + O(\|d^k\|^2)) \\ &\leq \kappa_1 \|A_k^T\| \|A_k\| ((1 - \eta_k)\|h^k\| + O(\|d^k\|^2)). \end{aligned} \tag{4.26}$$

Owing to $\|h^k\| = O(\|x^k - x^*\|) = O(\|d^k\|)$ and $(1 - \eta_k) = o(1)$, we can obtain from (4.26) that

$$\|A_k^T \tilde{d}^k\| = o(\|d^k\|),$$

and then complete the proof by (4.25). □

**Lemma 4.7** *Suppose that Assumption 4.2 and Assumption 4.5 hold. If* $\|(B_k - \nabla^2_{xx} L(x^*, \lambda^*))d\| = o(\|d\|)$ *at every* $d \in \Re^n$, $\sigma < 1/2$, $\|h^k + A_k^T d^k\| \leq (1 - \eta_k)\|h^k\|$, $\|h(x^k + d^k) + A_k^T \tilde{d}^k\| \leq (1 - \eta_k)\|h(x^k + d^k)\|$, $(1 - \eta_k) = o(1)$, *and*

$$v^k = O(\|d^k\|^2), \tag{4.27}$$

*then (4.6) and (4.7) hold for sufficiently large* $k$.

*Proof.* The **Condition (a)** and (4.27) imply that

$$\|d_p^k\| = O(\|d^k\|^2). \tag{4.28}$$

Thus, for sufficiently large $k$, there holds

$$\begin{aligned}(d^k)^T B_k d^k &= (d^k - d_p^k)^T B_k (d^k - d_p^k) + 2(d^k)^T B_k d_p^k - (d_p^k)^T B_k d_p^k \\ &\geq \tilde{\gamma}\|d^k - d_p^k\|^2 - O(\|d^k\|^3) \\ &\geq \hat{\gamma}\|d^k\|^2, \end{aligned} \tag{4.29}$$

where $\hat{\gamma} > 0$ is a constant and $\hat{\gamma} < \tilde{\gamma}$. Since $d^k$ is the solution of program (2.1)–(2.2), then one has

$$(g^k)^T d^k + (1/2)(d^k)^T B_k d^k \leq (g^k)^T d_p^k + (1/2)(d_p^k)^T B_k d_p^k.$$

By (4.28) and (4.29), we obtain that, for sufficiently large $k$,

$$(g^k)^T (d^k - d_p^k) \leq -\check{\gamma}\|d^k\|^2,$$

where $\check{\gamma}$ is a constant and $\check{\gamma} < \hat{\gamma}$. Therefore, by (4.28), as $\|d^k\| \to 0$, for sufficiently large $k$, there exists a constant $\tilde{\xi} > 0$ such that

$$(g^k)^T d^k \leq -\tilde{\xi}\|d^k\|^2. \tag{4.30}$$

By the setting of $v_{\max}^k$, without loss of generality, we assume that $v_{\max}^k \geq \epsilon\|d^k\|^2$ for some small $\epsilon > 0$. Owing to $v^k \leq \max\{(r_k+1)/2, 0.95\}v_{\max}^k$,

$$\begin{aligned} v(x^k + d^k + \tilde{d}^k) &= \|h(x^k + d^k + \tilde{d}^k)\| \\ &\leq \|h(x^k + d^k) + A_k^T \tilde{d}^k\| + o(\|d^k\|^2) \\ &\leq (1-\eta_k) v(x^k + d^k) + o(\|d^k\|^2), \end{aligned} \tag{4.31}$$

and $v(x^k + d^k) \leq \|h^k + A_k^T d^k\| + O(\|d^k\|^2) \leq (1-\eta_k)v^k + O(\|d^k\|^2)$, then

$$\begin{aligned} v(x^k + d^k + \tilde{d}^k) &\leq (1-\eta_k)^2 \max\{(r_k+1)/2, 0.95\}v_{\max}^k + o(\|d^k\|^2) \\ &\leq \max\{(r_k+1)/2, 0.95\}v_{\max}^k - 0.95\eta_k(2-\eta_k)\epsilon\|d^k\|^2 + o(\|d^k\|^2). \end{aligned}$$

Thus, the inequality (4.7) holds for sufficiently large $k$.

It follows from (4.21) and (4.27) that

$$v(x^k + d^k) = O(\|d^k\|^2). \tag{4.32}$$

Consequently, by (4.31), and due to $(1-\eta_k) = o(1)$,

$$v(x^k + d^k + \tilde{d}^k) = o(\|d^k\|^2). \tag{4.33}$$

Both (4.30) and (4.33) imply that for sufficiently large $k$

$$\sigma(g^k)^T d^k \leq -\xi_1 v(x^k + d^k + \tilde{d}^k).$$

Hence, in order to prove that (4.6) holds for sufficiently large $k$, we are left to prove that for sufficiently large $k$ there holds

$$f(x^k + d^k + \tilde{d}^k) - f^k \leq \sigma(g^k)^T d^k. \tag{4.34}$$

Noticing that $\|x^k + d^k + \tilde{d}^k - x^*\| = o(\|d^k\|)$, $\|x^k - x^*\| = O(\|d^k\|)$ and

$$\begin{aligned}
&L(x^k + d^k + \tilde{d}^k, \lambda^*) - L(x^k, \lambda^*) \\
&= L(x^k + d^k + \tilde{d}^k, \lambda^*) - L(x^*, \lambda^*) - (L(x^k, \lambda^*) - L(x^*, \lambda^*)) \\
&= -(1/2)(x^k - x^*)^T \nabla^2_{xx} L(x^*, \lambda^*)(x^k - x^*) + o(\|d^k\|^2),
\end{aligned}$$

we can deduce that

$$\begin{aligned}
&f(x^k + d^k + \tilde{d}^k) - f^k \\
&= L(x^k + d^k + \tilde{d}^k, \lambda^*) - L(x^k, \lambda^*) - (\lambda^*)^T (h(x^k + d^k + \tilde{d}^k) - h^k) \\
&= -(1/2)(x^k - x^*)^T \nabla^2_{xx} L(x^*, \lambda^*)(x^k - x^*) - (\lambda^*)^T (h(x^k + d^k + \tilde{d}^k) - h^k) + o(\|d^k\|^2) \\
&= -(1/2)(x^k - x^*)^T \nabla^2_{xx} L(x^*, \lambda^*)(x^k - x^*) - (\lambda^*)^T A_k^T (d^k + \tilde{d}^k) \\
&\quad -(1/2) \sum_{j=1}^m \lambda_j^* (d^k + \tilde{d}^k)^T \nabla^2 h_j^k (d^k + \tilde{d}^k) + o(\|d^k\|^2) \\
&= -(1/2)(x^k - x^*)^T \nabla^2_{xx} L(x^*, \lambda^*)(x^k - x^*) - \nabla_x L(x^k, \lambda^*)^T d^k - (\lambda^*)^T A_k^T \tilde{d}^k \\
&\quad +(g^k)^T d^k - (1/2) \sum_{j=1}^m \lambda_j^* (d^k + \tilde{d}^k)^T \nabla^2 h_j^k (d^k + \tilde{d}^k) + o(\|d^k\|^2) \\
&= -(1/2)(x^k - x^*)^T \nabla^2_{xx} L(x^*, \lambda^*) d^k - (1/2) \sum_{j=1}^m \lambda_j^* (d^k)^T \nabla^2 h_j^k d^k \\
&\quad -(\lambda^*)^T A_k^T \tilde{d}^k + (g^k)^T d^k + o(\|d^k\|^2).
\end{aligned}$$

Combining with the results $\|x^k + d^k - x^*\| = o(\|d^k\|)$, $\|\tilde{d}^k\| = o(\|d^k\|)$, and

$$\sum_{j=1}^m \lambda_j^* (d^k)^T \nabla^2 h_j^k d^k = -(\lambda^*)^T (A_k - A_*)^T d^k + o(\|d^k\|^2), \tag{4.35}$$

we can further deduce that

$$\begin{aligned}
&f(x^k + d^k + \tilde{d}^k) - f^k - \sigma(g^k)^T d^k \\
&= (1/2)(d^k)^T \nabla^2_{xx} L(x^*, \lambda^*) d^k - (1/2) \sum_{j=1}^m \lambda_j^* (d^k)^T \nabla^2 h_j^k d^k \\
&\quad +(1 - \sigma)(g^k)^T d^k - (\lambda^*)^T A_k^T \tilde{d}^k + o(\|d^k\|^2), \\
&= (1/2)(d^k)^T \nabla^2_{xx} L(x^*, \lambda^*) d^k + (1/2) \nabla_x L(x^k, \lambda^*)^T d^k - (1/2)(A_* \lambda^*)^T d^k \\
&\quad +(1/2 - \sigma)(g^k)^T d^k - (\lambda^*)^T A_k^T \tilde{d}^k + o(\|d^k\|^2) \\
&= (1/2 - \sigma)(g^k)^T d^k - (1/2)(\lambda^*)^T A_*^T d^k - (\lambda^*)^T A_k^T \tilde{d}^k + o(\|d^k\|^2).
\end{aligned} \tag{4.36}$$

Due to

$$
\begin{aligned}
&(1/2)(\lambda^*)^T A_*^T d^k + (\lambda^*)^T A_k^T \tilde{d}^k \\
&= (1/2)(\lambda^*)^T (A_* - A_k)^T d^k + (1/2)(\lambda^*)^T A_k^T (d^k + \tilde{d}^k) + (1/2)(\lambda^*)^T A_k^T \tilde{d}^k + o(\|d^k\|^2) \\
&= (1/4)(\lambda^*)^T (A_* - A_k)^T d^k + (\lambda^*)^T h(x^k + d^k + \tilde{d}^k) - (1/2)(\lambda^*)^T (h^k + h(x^k + d^k)) \\
&\quad + o(\|d^k\|^2) \\
&= (1/4)(\lambda^*)^T (A_* - A_k)^T d^k - (1/2)(\lambda^*)^T (h^k + h(x^k + d^k)) + o(\|d^k\|^2) \\
&= (1/4)(\lambda^*)^T (A_* - A_k)^T d^k - (1/2)(\lambda^*)^T (h^k + A_k^T d^k) - (1/2)(\lambda^*)^T (h(x^k + d^k) + A_k^T \tilde{d}^k) \\
&\quad + (1/2)(\lambda^*)^T A_k^T (d^k + \tilde{d}^k) + o(\|d^k\|^2) \\
&= (1/4)(\lambda^*)^T (A_* - A_k)^T d^k + (1/2)(\lambda^*)^T A_k^T (d^k + \tilde{d}^k) + o(\|d^k\|^2) \\
&= (1/4)(\lambda^*)^T A_*^T d^k + (1/4)(\lambda^*)^T A_k^T d^k + (1/2)(\lambda^*)^T A_k^T \tilde{d}^k + o(\|d^k\|^2),
\end{aligned}
$$

where the second equality is obtained by (4.35), the third by (4.33), the fifth by (4.27) and (4.32), it follows that

$$
(1/2)(\lambda^*)^T A_*^T d^k + (\lambda^*)^T A_k^T \tilde{d}^k = (1/2)(\lambda^*)^T A_k^T d^k + o(\|d^k\|^2).
$$

Hence, by (4.36), we have

$$
\begin{aligned}
&f(x^k + d^k + \tilde{d}^k) - f^k - \sigma (g^k)^T d^k \\
&= (1/2 - \sigma)(g^k)^T d^k - (1/2)(\lambda^*)^T A_k^T d^k + o(\|d^k\|^2) \\
&= (1/2 - \sigma)(g^k)^T d^k - (1/2)(\lambda^*)^T A_k^T d_p^k + o(\|d^k\|^2) \\
&= (1/2 - \sigma)(g^k)^T d^k + (1/2)(g^k)^T d_p^k + o(\|d^k\|^2),
\end{aligned}
$$

where the second equality follows from (2.2) and the third from $(g^k + A_k \lambda^*)^T d_p^k = o(\|d^k\|^2)$. Therefore, (4.34) follows from (4.30) due to $\sigma < 1/2$ and $\|d_p^k\| = O(\|d^k\|^2)$ (see (4.28)). □

Making use of the results in Lemma 4.4 and Lemma 4.7, we can conclude that the full superlinearly convergent step will be accepted by Algorithm 4.1 for sufficiently large $k$.

**Theorem 4.8** *Suppose that Assumption 4.2 and Assumption 4.5 hold. If* $\|(B_k - \nabla_{xx}^2 L(x^*, \lambda^*))d\| = o(\|d\|)$ *at every* $d \in \Re^n$, $\sigma < 1/2$, $\|h^k + A_k^T d^k\| \le (1 - \eta_k)\|h^k\|$, $\|h(x^k + d^k) + A_k^T \tilde{d}^k\| \le (1 - \eta_k)\|h(x^k + d^k)\|$, *and* $(1 - \eta_k) = o(1)$, *then either* $x^{k+1} = x^k + d^k$ *or* $x^{k+1} = x^k + d^k + \tilde{d}^k$ *for sufficiently large* $k$.

*Proof.* For sufficiently large $k$, if either (4.1) or (4.3) is satisfied, then $x^{k+1} = x^k + d^k$; otherwise, (4.3) does not hold, thus by Lemma 4.4 and Lemma 4.7, (4.6) and (4.7) will then be satisfied for sufficiently large $k$, which give rise to $x^{k+1} = x^k + d^k + \tilde{d}^k$ by Algorithm 4.1. □

## 5. Numerical experiments

We have implemented Algorithm 2.2 in MATLAB, run with version R2008a. The numerical test were conducted on a Lenovo laptop with the LINUX operating system (Fedora 11). A set of 60 small- and medium-size test problems from the CUTE collection [3] were solved. For comparison, these problems were also solved with LANCELOT [10], a state-of-the-art solver for nonlinear constrained optimization problems, in which the BFGS approximate second-order derivative was used.

**5.1. Solving the subproblems.** We compute the null space matrix $W_k$ of $A_k^T$ directly by the MATLAB's null space routine, which generates an orthonormal basis for the null space of $A_k^T$ obtained from the singular value decomposition. The solution of subproblem (2.1)-(2.2) is obtained by forming the reduced Hessian explicitly and using the MATLAB's routine of bi-conjugate gradients method with preconditioner generated by the sparse incomplete Cholesky-Infinity factorization.

Intentionally, we compute $d_p^k$ to satisfy the **Conditions (a)–(b)**. Firstly, we solve the equation $(A_k^T A_k)p = h^k$ by the bi-conjugate gradients method to get a solution $p^k$ and set $d_n^k = -A_k p^k$. We then take $d_p^k = d_n^k$ provided the conditions $\|h^k + A_k^T d_n^k\| \le \min\{0.5, \|h^k\|\}\|h^k\|$ and $\|d_n^k\| \le \kappa \max\{\theta_k, 1\}\|A_k h^k\|$ are satisfied, where $\kappa = 10^4$ in our implementation and $\theta_k$ is the same as that in Lemma 2.1; otherwise, we compute a Cauchy step by the formula $d_c^k = -\theta_k A_k h^k$ and calculate $(\mu_k, \nu_k)$ which minimizes $\|h^k + A_k^T(\mu d_n^k + \nu d_c^k)\|$, and if both $\mu_k$ and $\nu_k$ are finite (which are decided by the MATLAB function *isfinite*), we set $d_p^k = \mu_k d_n^k + \nu_k d_c^k$, else we take $d_p^k = d_c^k$.

**5.2. Numerical results.** In Algorithm 2.2, $B_k$ is updated by the Powell's damped BFGS update procedure [22, 26], where the estimate $\lambda^k$ of the Lagrangian multipliers are given by MATLAB's LSQR method. The parameters are selected as follows: $\xi_1 = 10^{-10}$, $\xi_2 = 10^{-4}$, $\sigma = 0.01$. The step-size is decided by the Armijo line search procedure with $\alpha_k = \tau^\ell$ and $\tau = 0.6, \ell = 0, 1, \ldots$. The algorithm is terminated with $\epsilon = 10^{-5}$.

We explain how the problems in CUTE were selected in our comparisons. In Table 1 and Table 2, we report the computational results for those problems where the difference between the optimal function value achieved by our algorithm and that obtained by LANCELOT was within the termination tolerance. Some other problems were not included since either our algorithm found a solution with different objective value from that derived by LANCELOT (that is, the difference between them was out of the termination tolerance, such as for problems BT2, HS47, MWRIGHT), or either our algorithm or LANCELOT did not find a solution before reaching the restriction of number 1000 of function evaluations (such as problems HEART6, HEART8, HS56, HS111LNP, HYDCAR6, HYDCAR20 for which LANCELOT reached the restriction of number 1000 of function evaluations before the termination for a solution, and problems BT7, BYRDSPHR, HATFLDF, HS27, POWELLSQ for which our algorithm reached the restriction of number 1000 of function evaluations before the termination for a solution).

The number of $f, h$ evaluations and the number of $g, A$ evaluations for testing problems are listed in Tables 1 and 2, where the column "Algorithm 2.2" shows the results for Algorithm 2.2 and the column "LANCELOT" shows the results for LANCELOT (Version 27/02/2001). The specification file for LANCELOT was changed only in that the setting on the use of exact Hessian information was replaced by the use of BFGS approximate Hessian.

Table 1: Results for CUTE problems using approximate Hessians, Part 1

| | Algorithm 2.2 | | LANCELOT | | Problem Dim. | |
|---|---|---|---|---|---|---|
| Problem | $f, h$ | $g, A$ | $f, h$ | $g, A$ | $n$ | $m$ |
| AIRCRFTA | 3 | 3 | 5 | 5 | 8 | 5 |
| BDVALUE | 3 | 2 | 2 | 2 | 502 | 500 |
| BDVALUES | 10 | 10 | 20 | 20 | 12 | 10 |
| BOOTH | 2 | 2 | 4 | 4 | 2 | 2 |
| BRATU2D | 2 | 2 | 4 | 4 | 484 | 400 |
| BRATU2DT | 2 | 2 | 8 | 8 | 484 | 400 |
| BRATU3D | 2 | 2 | 5 | 5 | 1000 | 512 |
| BT1 | 11 | 8 | 57 | 47 | 2 | 1 |
| BT3 | 8 | 8 | 15 | 15 | 5 | 3 |
| BT4 | 14 | 14 | 27 | 26 | 3 | 2 |
| BT5 | 9 | 9 | 67 | 43 | 3 | 2 |
| BT6 | 30 | 29 | 51 | 39 | 5 | 2 |
| BT8 | 11 | 11 | 27 | 25 | 5 | 2 |
| BT9 | 57 | 41 | 23 | 23 | 4 | 2 |
| BT10 | 8 | 8 | 21 | 21 | 2 | 3 |
| BT11 | 13 | 13 | 23 | 20 | 5 | 3 |
| BT12 | 9 | 8 | 23 | 19 | 5 | 4 |
| CBRATU2D | 2 | 2 | 4 | 4 | 512 | 392 |
| CBRATU3D | 3 | 3 | 5 | 5 | 686 | 250 |
| CLUSTER | 8 | 8 | 13 | 10 | 2 | 2 |
| CUBENE | 12 | 5 | 42 | 34 | 2 | 2 |
| DECONVNE | 4 | 4 | 28 | 23 | 61 | 40 |
| GENHS28 | 9 | 9 | 10 | 10 | 10 | 8 |
| GOTTFR | 9 | 6 | 12 | 11 | 2 | 2 |
| HATFLDG | 25 | 7 | 24 | 20 | 25 | 25 |
| HIMMELBA | 2 | 2 | 3 | 3 | 2 | 2 |
| HIMMELBC | 7 | 6 | 9 | 8 | 2 | 2 |
| HIMMELBE | 3 | 3 | 4 | 4 | 3 | 3 |
| HS6 | 14 | 11 | 58 | 42 | 2 | 1 |
| HS7 | 12 | 12 | 24 | 19 | 2 | 1 |
| HS8 | 6 | 5 | 11 | 10 | 2 | 2 |
| HS9 | 7 | 7 | 11 | 11 | 2 | 1 |
| HS26 | 36 | 26 | 33 | 31 | 3 | 1 |
| HS28 | 10 | 9 | 4 | 4 | 3 | 1 |
| HS39 | 57 | 41 | 674 | 630 | 4 | 2 |
| HS40 | 7 | 7 | 15 | 14 | 4 | 3 |
| HS42 | 11 | 9 | 13 | 13 | 4 | 2 |
| HS46 | 29 | 27 | 28 | 25 | 5 | 2 |
| HS48 | 13 | 10 | 4 | 4 | 5 | 2 |
| HS49 | 27 | 22 | 25 | 25 | 5 | 2 |
| HS50 | 25 | 15 | 19 | 19 | 5 | 3 |
| HS51 | 10 | 9 | 3 | 3 | 5 | 3 |

Table 2: Results for CUTE problems using approximate Hessians, Part 2

| | Algorithm 2.2 | | LANCELOT | | Problem Dim. | |
|---|---|---|---|---|---|---|
| Problem | $f, c$ | $g, A$ | $f, c$ | $g, A$ | $n$ | $m$ |
| HS52 | 8 | 7 | 11 | 11 | 5 | 3 |
| HS61 | 13 | 11 | 18 | 17 | 3 | 2 |
| HS77 | 29 | 26 | 35 | 30 | 5 | 2 |
| HS78 | 9 | 9 | 26 | 15 | 5 | 3 |
| HS79 | 13 | 13 | 12 | 12 | 5 | 3 |
| HS100LNP | 79 | 35 | 510 | 468 | 7 | 2 |
| HYPCIR | 6 | 5 | 7 | 7 | 2 | 2 |
| INTEGREQ | 2 | 2 | 3 | 3 | 102 | 100 |
| MARATOS | 5 | 5 | 9 | 9 | 2 | 1 |
| METHANB8 | 3 | 3 | 348 | 342 | 31 | 31 |
| ORTHREGB | 7 | 7 | 140 | 116 | 27 | 6 |
| POWELLBS | 26 | 16 | 48 | 42 | 2 | 2 |
| RECIPE | 12 | 12 | 43 | 37 | 2 | 2 |
| RSNBRNE | 36 | 10 | 34 | 30 | 2 | 2 |
| S316-322 | 9 | 8 | 24 | 24 | 2 | 1 |
| SINVALNE | 35 | 10 | 37 | 30 | 2 | 2 |
| TRIGGER | 8 | 8 | 20 | 18 | 7 | 6 |
| ZANGWIL3 | 2 | 2 | 8 | 8 | 3 | 3 |

Although our MATLAB implementation uses MATLAB's routines simply, the number of function and gradient evaluations for most problems are very satisfactory. The fast local convergence during the numerical solution is also observed, which is consistent with our local convergence theory. These results show that the SQP method without a penalty function or a filter is robust and efficient. Moreover, it is believed that further improvements can be achieved by using advanced techniques for, e.g., the computations of $W_k$ and $d_p^k$.

**Acknowledgements.** The authors are grateful to the associate editor and two anonymous referees for their very careful reading and their very helpful comments on two earlier versions of the paper, and to the associate editor for his proposal of the terminating conditions. They also appreciate Dr. Zhouhong Wang and Ms Caixia Kou for their help in preparing the numerical experiments.

# References

[1] R. Andreani, E.G. Birgin, J.M. Martinez, and M.L. Schuverdt, *Augmented Lagrangian methods under the constant positive linear dependence constraint qualification,* Math. Program., 111(2008), 5–32.

[2] P. Armand and J. Benoist, *A local convergence property of primal-dual methods for nonlinear*

*programming,* Math. Program., 115(2008), 199–222.

[3] I. Bongartz, A.R. Conn, N.I.M. Gould and P.L. Toint, *CUTE: Constrained and Unconstrained Testing Environment,* ACM Tran. Math. Software, 21(1995), 123–160.

[4] J.V. Burke and S.P. Han, *A robust sequential quadratic programming method,* Math. Program., 43(1989), 277–303.

[5] R.H. Byrd, J.C. Gilbert and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming,* Math. Program., 89(2000), 149–185.

[6] R.H. Byrd, G. Liu and J. Nocedal, *On the local behaviour of an interior point method for nonlinear programming,* In D.F. Griffiths and D.J. Higham, editors, Numerical Analysis 1997, pp. 37-56, Addison-Wesley Longman, Reading, MA, 1997.

[7] R.H. Byrd, M. Marazzi and J. Nocedal, *On the convergence of Newton iterations to non-stationary points,* Math. Program., 99(2004), 127–148.

[8] L.F. Chen and D. Goldfarb, *Interior-point $\ell_2$-penalty methods for nonlinear programming with strong global convergence properties,* Math. Program., 108(2006), 1–36.

[9] C.M. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps,* Report NA/199, Department of Mathematics, Dundee University, Dundee, Scotland, 2001.

[10] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A),* Springer-Verlag, 1992.

[11] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *Trust-Region Methods,* SIAM, Philadelphia, PA, USA, 2000.

[12] R. Fletcher, N. Gould, S. Leyffer, Ph.L. Toint and A. Wächter, *Global convergence of a trust-region SQP-filter allgorithm for general nonlinear programming,* SIAM J. Optim., 13(2002), 635–659.

[13] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function,* Math. Program., 91(2002), 239–269.

[14] R. Fletcher, S. Leyffer and Ph.L. Toint, *On the global convergence of a filter-SQP algorithm,* SIAM J. Optim., 13(2002), 44–59.

[15] A. Forsgren, *Inertia-controlling factorizations for optimization algorithms,* Appl. Numer. Math., 43(2002), 91–107.

[16] N.I.M. Gould, *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem,* Math. Program. 32(1985), 90–99.

[17] N.I.M. Gould and Ph.L. Toint, *Nonlinear programming without a penalty function or a filter,* Math. Program., 122(2009), 155–196.

[18] I. Griva, D.F. Shanno and R.J. Vanderbei, *Convergence analysis of a primal-dual interior-point method for nonlinear programming,* Technical Report ORFE-04-07, Department of ORFE, Princeton University, Princeton, NJ 08544, 2004.

[19] X.-W. Liu and J. Sun, *A robust primal-dual interior point algorithm for nonlinear programs,* SIAM J. Optim., 14(2004), 1163–1186.

[20] X.-W. Liu and Y.-X. Yuan, *A robust algorithm for optimization with general equality and inequality constraints,* SIAM J. Sci. Comput., 22(2000), 517–534.

[21] X.-W. Liu and Y.-X. Yuan, *A null-space primal-dual interior-point algorithm for nonlinear optimization with nice convergence properties,* Math. Program., 125(2010), 163–193.

[22] J. Nocedal and S. Wright, *Numerical Optimization,* Springer-Verlag New York, Inc., 1999.

[23] M.J.D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization,* Math. Program., 49(1991), 189–211.

[24] A.A. Ribeiro, E.W. Karas, C.C. Gonzaga, *Global convergence of filter methods for nonlinear programming,* the 19th ISMP, Brazil.

[25] M. Solodov, *Global convergence of an SQP method without boundedness assumptions on any of the iterative sequences,* Math. Program., 118(2009), 1–12.

[26] W.Y. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming,* Springer, New York, 2006.

[27] M. Ulbrich and S. Ulbrich, *Non-monotone trust region methods for nonlinear equality constrained optimization without a penalty function,* Math. Program., 95(2003), 103–135.

[28] M. Ulbrich, S. Ulbrich and L.N. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming,* Math. Program., 100(2004), 379–410.

[29] A. Wächter and L.T. Biegler, *Line search filter methods for nonlinear programming: Motivation and global convergence,* SIAM J. Optim., 16(2005), 1–31.

[30] A. Wächter and L.T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,* Math. Program., 106(2006), 25–57.

[31] H. Yamashita and H. Yabe, *Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization,* Math. Program., 75(1996), 377–397.

[32] H. Yamashita and H. Yabe, *A globally and superlinearly convergent trust-region SQP method without a penalty function for nonlinearly constrained optimization,* revised 2007.

[33] Y. Yuan, *On the convergence of a new trust region algorithm,* Numer. Math., 70(1995), 515–539.