Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 971468, 15 pages
doi:10.1155/2012/971468

*Research Article*

# A Nonmonotone Line Search Filter Algorithm for the System of Nonlinear Equations

**Zhong Jin[1] and Yuqing Wang[2]**

[1] *Department of Mathematics, Shanghai Maritime University, Shanghai 201306, China*
[2] *Xianda College of Economics and Humanities, Shanghai International Studies University, Shanghai 200083, China*

Correspondence should be addressed to Zhong Jin, asdjz1234@163.com

Received 9 June 2012; Accepted 23 June 2012

Academic Editor: Wei-Chiang Hong



We present a new iterative method based on the line search filter method with the nonmonotone strategy to solve the system of nonlinear equations. The equations are divided into two groups; some equations are treated as constraints and the others act as the objective function, and the two groups are just updated at the iterations where it is needed indeed. We employ the nonmonotone idea to the sufficient reduction conditions and filter technique which leads to a flexibility and acceptance behavior comparable to monotone methods. The new algorithm is shown to be globally convergent and numerical experiments demonstrate its effectiveness.

## 1. Introduction

We consider the following system of nonlinear equations:

$$c_i(x) = 0, \quad i = 1, 2, \ldots, m, \tag{1.1}$$

where each $c_i : R^n \rightarrow R$ $(i = 1, 2, \ldots, m)$ is a twice continuously differentiable function. It is one of the most basic problems in mathematics and has lots of applications in many scientific fields such as physics, chemistry, and economics.

In the context of solving nonlinear equations, a well-known method is the Newton method, which is known to exhibit local and second order convergence near a regular solution, but its global behavior is unpredictable. To improve the global properties, some important algorithms [1] for nonlinear equations proceed by minimizing a least square problem:

$$\min h(x) = c(x)^T c(x), \tag{1.2}$$

which can be also handled by the Newton method, while Powell [2] gives a counterexample to show a dissatisfactory fact that the iterates generated by the above least square problem may converge to a nonstationary point of $h(x)$.

However, as we all know, there are several difficulties in utilizing the penalty functions as a merit function to test the acceptability of the iterates. Hence, the filter, a new concept first introduced by Fletcher and Leyffer [3] for constrained nonlinear optimization problems in a sequential quadratic programming (SQP) trust-region algorithm, replaces the merit fuctions avoiding the penalty parameter estimation and the difficulties related to the nondifferentiability. Furthermore, Fletcher et al. [4, 5] give the global convergence of the trust-region filter-SQP method, then Ulbrich [6] gets its superlinear local convergence. Consequently, filter method has been actually applied in many optimization techniques, for instance the pattern search method [7], the SLP method [8], the interior method [9], the bundle approaches [10, 11], and so on. Also combined with the trust-region search technique, Gould et al. extended the filter method to the system of nonlinear equations and nonlinear least squares in [12], and to the unconstrained optimization problem with multidimensional filter technique in [13]. In addition, Wächter and Biegler [14, 15] presented line search filter methods for nonlinear equality-constrained programming and the global and local convergence were given.

In fact, filter method exhibits a certain degree of nonmonotonicity. The idea of nonmonotone technique can be traced back to Grippo et al. [16] in 1986, combined with the line search strategy. Due to its excellent numerical exhibition, many nonmonotone techniques have been developed in recent years, for example [17, 18]. Especially in [17], a nonmonotone line search multidimensional filter-SQP method for general nonlinear programming is presented based on the Wächter and Biegler methods [14, 15].

Recently, some other ways were given to attack the problem (1.1) (see [19–23]). There are two common features in these papers; one is the filter approach is utilized, and the other is that the system of nonlinear equations is transformed into a constrained nonlinear programming problem and the equations are divided into two groups; some equations are treated as constraints and the others act as the objective function. And two groups of equations are updated at every iteration in those methods. For instance combined with the filter line search technique [14, 15], the system of nonlinear equations in [23] becomes the following optimization problem with equality constraints:

$$\begin{aligned} \min \quad & \sum_{i\in S_1} c_i^2(x) \\ \text{s.t.} \quad & c_j(x) = 0, \quad j \in S_2. \end{aligned} \tag{1.3}$$

The choice of two sets $S_1$ and $S_2$ are given as follows: for some positive constant $n_0 > 0$, it is defined that $c_{i_1}^2(x_k) \geq c_{i_2}^2(x_k) \geq \cdots \geq c_{i_m}^2(x_k)$, then $S_1 = \{i_k \mid k \leq n_0\}$ and $S_2 = \{i_k \mid k \geq n_0 + 1\}$.

In this paper we present an algorithm to solve the system of nonlinear equations, combining the nonmonotone technique and line search filter method. We also divide the equations into two groups; one contains the equations that are treated as equality constraints and the square of other equations is regarded as objective function. But different from those methods in [19–23], we just update the two groups at the iterations where it is needed indeed, which can make the scale of the calculation decrease in a certain degree. Another merit of our paper is to employ the nonmonotone idea to the sufficient reduction conditions and filter which leads to a flexibility and acceptance behavior comparable to monotone methods. Moreover, in our algorithm two groups of equations cannot be changed after an f-type iteration, thus in the case that $|\mathcal{A}| < \infty$, the two groups are fixed after finite number of iterations. And the filter should not be updated after an f-type iteration, so naturally the global convergence is discussed, respectively, according to whether the number of updated filter is infinite or not. Furthermore, the global convergent property is induced under some reasonable conditions. In the end, numerical experiments show that the method in this paper is effective.

The paper is outlined as follows. In Section 2, we describe and analyze the nonmonotone line search filter method. In Section 3 we prove the global convergence of the proposed algorithm. Finally, some numerical tests are given in Section 4.

## 2. A Nonmonotone Line Search Filter Algorithm

Throughout this paper, we use the notations $m_k(x) = \|c_{S_1}(x)\|_2^2 = \sum_{i\in S_1} c_i^2(x)$ and $\theta_k(x) = \|c_{S_2}(x)\|_2^2 = \sum_{i\in S_2} c_i^2(x)$. In addition, we denote the set of indices of those iterations in which the filter has been augmented by $\mathcal{A} \subseteq \mathbb{N}$.

The linearization of the KKT condition of (1.3) at the $k$th iteration $x_k$ is as follows:

$$\begin{pmatrix} B_k & A_{S_2}^k \\ \left(A_{S_2}^k\right)^T & 0 \end{pmatrix} \begin{pmatrix} s_k \\ \lambda_k^+ \end{pmatrix} = -\begin{pmatrix} g_k \\ c_{S_2}^k \end{pmatrix}, \tag{2.1}$$

where $B_k$ is the Hessian or approximate Hessian matrix of $L(x,\lambda) = m_k(x) + \lambda^T c_{S_2}(x)$, $A_{S_2}^k = \nabla c_{S_2}(x_k)$ and $g(x_k) = \nabla m_k(x_k)$. Then the iterate formation is $x_k(\alpha_{k,l}) = x_k + \alpha_{k,l} s_k$, where $s_k$ is the solution of (2.1) and $\alpha_{k,l} \in (0,1]$ is a step size chosen by line search.

Now we describe the nonmonotone Armijo rule. Let $M$ be a nonnegative integer. For each $k$, let $m(k)$ satisfy $m(0) = 1, 0 \le m(k) \le \min\{m(k-1)+1, M\}$ for $k \ge 1$. For fixed constants $\gamma_m, \gamma_\theta \in (0,1)$, we might consider a trial point to be acceptable, if it leads to sufficient progress toward either goal, that is, if

$$\begin{gathered} \theta_k(x_k(\alpha_{k,l})) \le (1-\gamma_\theta)\max\left\{\theta_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr}\theta_{k-r}(x_{k-r})\right\} \\ \text{or} \quad m_k(x_k(\alpha_{k,l})) \le \max\left\{m_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} m_{k-r}(x_{k-r})\right\} - \gamma_m\theta_k(x_k), \end{gathered} \tag{2.2}$$

where $\lambda_{kr} \in (0,1)$, $\sum_{r=0}^{m(k)-1} \lambda_{kr} = 1$.

For the convenience we set $\overline{m}(x_k) = \max\{m_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} m_{k-r}(x_{k-r})\}$, and $\overline{\theta}(x_k) = \max\{\theta_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr}\theta_{k-r}(x_{k-r})\}$. In order to avoid the case of convergence to a feasible but nonoptimal point, we consider the following switching condition:

$$g_k^T s_k < -\xi s_k^T B_k s_k, \qquad -\alpha_{k,l} g_k^T s_k > [\theta_k(x_k)]^{s_\theta}, \tag{2.3}$$

with $\xi \in (0,1]$, $s_\theta \in (0,1)$. If the switching condition holds, the trial point $x_k(\alpha_{k,l})$ has to satisfy the Armijo nonmonotone reduction condition,

$$m_k(x_k(\alpha_{k,l})) \leq \overline{m}(x_k) + \tau_3 \alpha_{k,l} g_k^T s_k, \tag{2.4}$$

where $\tau_3 \in (0, 1/2)$ is a fixed constant.

To ensure the algorithm cannot cycle, it maintains a filter, a "taboo region" $\mathcal{F}_k \subseteq [0,\infty] \times [0,\infty]$ for each iteration $k$. The filter contains those combinations of constraint violation value $\theta$ and the objective function value $m$, that are prohibited for a successful trial point in iteration $k$. During the line search, a trial point $x_k(\alpha_{k,l})$ is rejected, if $(\overline{\theta}(x_k(\alpha_{k,l})), \overline{m}(x_k(\alpha_{k,l}))) \in \mathcal{F}_k$. We then say that the trial point is not acceptable to the current filter, which is also called $x_k(\alpha_{k,l}) \in \mathcal{F}_k$.

If a trial point $x_k(\alpha_{k,l}) \notin \mathcal{F}_k$ satisfies the switching condition (2.3) and the reduction condition (2.4), then this trial point is called an f-type point, and accordingly this iteration is called an f-type iteration. An f-type point should be accepted as $x_{k+1}$ with no updating of the filter, that is

$$\mathcal{F}_{k+1} = \mathcal{F}_k. \tag{2.5}$$

While if a trial point $x_k(\alpha_{k,l}) \notin \mathcal{F}_k$ does not satisfy the switching condition (2.3), but this trial point satisfies (2.2), we call it an h-type point, or accordingly an h-type iteration. An h-type point should be accepted as $x_{k+1}$ with updating of the filter, that is

$$\mathcal{F}_{k+1} = \mathcal{F}_k \bigcup \left\{ (\theta, m) \in R^2 : \theta \geq (1-\gamma_\theta)\overline{\theta}(x_k), m \geq \overline{m}(x_k) - \gamma_m \theta_k(x_k) \right\}. \tag{2.6}$$

In some cases it is not possible to find a trial step size that satisfies the above criteria. We approximate a minimum desired step size using linear models of the involved functions. For this, we define

$$\alpha_k^{\min} = \begin{cases} \min\left\{ 1 - \dfrac{(1-\gamma_\theta)\overline{\theta}(x_k)}{\theta_k(x_k)}, \dfrac{\overline{m}(x_k) - m_k(x_k) - \gamma_m \theta_k(x_k)}{g_k^T s_k}, \dfrac{[\theta_k(x_k)]^{s_\theta}}{-g_k^T s_k} \right\}, & \\ & \text{if } g_k^T s_k < -\xi s_k^T B_k s_k, \\ 1 - \dfrac{(1-\gamma_\theta)\overline{\theta}(x_k)}{\theta_k(x_k)}, & \text{otherwise.} \end{cases} \tag{2.7}$$

If the nonmonotone line search encounters a trial step size with $\alpha_{k,l} < \alpha_k^{\min}$, the algorithm reverts to a feasibility restoration phase. Here, we try to find a new iterate which is acceptable to the current filter and for which (2.2) holds, by reducing the constraint violation with some iterative method.

The corresponding algorithm can be written as follows.

*Algorithm 2.1.* *Step* 1. Initialization: choose an initial guess $x_0$, $\rho_1, \rho_2 \in (0,1)$, $\rho_1 < \rho_2$, and $\epsilon > 0$. Compute $g_0$, $c_i(x_0)$, $S_1^0$, $S_2^0$, and $A_k$ for $i \in S_2^0$. Set $M > 0$, $m(0) = 1$, $k = 0$, and $\mathcal{F}_0 = \emptyset$.

*Step* 2. If $\|c(x_k)\| \leq \epsilon$ then stop.

*Step* 3. Compute (2.1) to obtain $s_k$. If there exists no solution to (2.1), go to Step 8. If $\|s_k\| \leq \epsilon$ then stop.

*Step* 4. Use nonmonotone line search. Set $l = 0$ and $\alpha_{k,l} = 1$.

*Step* 4.1. If $\alpha_{k,l} < \alpha_k^{\min}$, where the $\alpha_k^{\min}$ is obtained by (2.7), go to Step 8. Otherwise we get $x_k(\alpha_{k,l}) = x_k + \alpha_{k,l} s_k$. If $x_k(\alpha_{k,l}) \in \mathcal{F}_k$, go to Step 4.3.

*Step* 4.2. Check sufficient decrease with respect to current iterate.

*Step* 4.2.1. If the switching condition (2.3) and the nonmonotone reduction condition (2.4) hold, set $\mathcal{F}_{k+1} = \mathcal{F}_k$ and go to Step 5. While only the switching condition (2.3) are satisfied, go to Step 4.3.

*Step* 4.2.2. The switching conditions (2.3) are not satisfied. If the nonmonotone filter condition (2.2) holds, set $x_{k+1} = x_k + \alpha_{k,l} s_k$, augment the filter using (2.6) and go to Step 6. Otherwise, go to Step 4.3.

*Step* 4.3. Choose $\alpha_{k,l+1} \in [\rho_1 \alpha_{k,l}, \rho_2 \alpha_{k,l}]$. Let $l = l + 1$ and go to Step 4.1.

*Step* 5. Set $x_{k+1} = x_k + \alpha_{k,l} s_k$, $S_1^{k+1} = S_1^k$ and $S_2^{k+1} = S_2^k$. Go to Step 7.

*Step* 6. Compute $S_1^{k+1}$ and $S_2^{k+1}$ by (1.3). If $(\overline{\theta}_{k+1}(x_{k+1}), \overline{m}_{k+1}(x_{k+1})) \in \mathcal{F}_{k+1}$, set $S_1^{k+1} = S_1^k$ and $S_2^{k+1} = S_2^k$.

*Step* 7. Compute $g_{k+1}$, $B_{k+1}$, $A_{k+1}$ and $m(k+1) = \min\{m(k)+1, M\}$. Let $k = k + 1$ and go to Step 2.

*Step* 8 (restoration stage). Find $x_k^r = x_k + \alpha_k^r s_k^r$ such that $x_k^r$ is acceptable to $x_k$ and $(\theta_k(x_k^r), m_k(x_k^r)) \notin \mathcal{F}_k$. Set $x_{k+1} = x_k^r$ and augment the filter by (2.6). Let $k = k + 1$, $m(k) = 1$ and go to Step 2.

In a restoration algorithm, the infeasibility is reduced and it is, therefore, desired to decrease the value of $\theta_k(x)$. The direct way is to utilize the Newton method or the similar ways to attack $\theta_k(x + s) = 0$. We now give the restoration algorithm.

*Restoration Algorithm*

*Step R1.* Let $x_k^0 = x_k$, $H_0 = E_n$, $\Delta_k^0 = \Delta^0$, $g_\theta = \nabla\theta_k(x)$, $j = 0$, $\eta_1 = 0.25$, $\eta_2 = 0.75$.

*Step R2*. If $x_k^j$ is acceptable to $x_k$ and $(\theta_k(x_k^r), m_k(x_k^r)) \notin \mathcal{F}_k$, then let $x_k^r = x_k^j$ and stop.

*Step R3*. Compute

$$\min g_\theta^T s + \frac{1}{2} s^T H_j s \quad \text{s.t. } \|s\| \leq \Delta_k^j \tag{2.8}$$

to get $s_k^j$. Let $r_k^j = (\theta_k(x_k^j) - \theta_k(x_k^j + s_k^j)) / (-g_\theta^T s_k^j - (1/2) {s_k^j}^T H_j s_k^j)$.

*Step R4.* If $r_k^j \le \eta_1$, set $\Delta_k^{j+1} = (1/2)\Delta_k^j$; If $r_k^j \ge \eta_2$, set $\Delta_k^{j+1} = 2\Delta_k^j$; otherwise, $\Delta_k^{j+1} = \Delta_k^j$. Let $x_k^{j+1} = x_k^j + s_k^j$, $H_j$ be updated to $H_{j+1}$, $j = j + 1$ and go to Step R2.

The above restoration algorithm is an SQP method for $\theta_k(x + s) = 0$. Of course, there are other restoration algorithms, such as the Newton method, interior point restoration algorithm, SLP restoration algorithm, and so on.

## 3. Global Convergence of Algorithm

In this section, we present a proof of global convergence of Algorithm 2.1. We first state the following assumptions in technical terms.

*Assumptions.* (A1) All points $x^k$ that are sampled by algorithm lie in a nonempty closed and bounded set $X$.

(A2) The functions $c_i(x)$, $j = 1, 2, \ldots, m$ are all twice continuously differentiable on an open set containing $X$.

(A3) There exist two constants $b \ge a > 0$ such that the matrices sequence $\{B_k\}$ satisfies $a\|s\|^2 \le s^T B_k s \le b\|s\|^2$ for all $k$ and $s \in R^n$.

(A4) $(A_{s_2}^k)^T$ has full column rank and $\|s_k\| \le \gamma_s$ for all $k$ with a positive constant $\gamma_s$.

In the remainder of this section, we will not consider the case where Algorithm 2.1 terminates successfully in Step 2, since in this situation the global convergence is trivial.

**Lemma 3.1.** *Under Assumption A1, there exists the solution to* (2.1) *with exact (or inexact) line search which satisfies the following descent conditions:*

$$|\theta_k(x_k + \alpha s_k) - (1 - 2\alpha)\theta_k(x_k)| \le \tau_1 \alpha^2 \|s_k\|^2, \tag{3.1}$$

$$\left| m_k(x_k + \alpha s_k) - m_k(x_k) - \alpha g_k^T s_k \right| \le \tau_2 \alpha^2 \|s_k\|^2, \tag{3.2}$$

*where $\alpha \in (0, 1)$, $\tau_1$ and $\tau_2$ are all positive constants independent of $k$.*

*Proof.* By virtue of the Taylor expansion of $c_i^2(x_k + \alpha s_k)$ with $i \in S_2$, we obtain

$$\begin{aligned}
&\left| c_i^2(x_k + \alpha s_k) - c_i^2(x_k) - 2\alpha c_i(x_k) \nabla c_i(x_k)^T s_k \right| \\
&\quad = \left| c_i^2(x_k + \alpha s_k) - c_i^2(x_k) - 2c_i(x_k) \nabla c_i(x_k)^T (\alpha s_k) \right| \\
&\quad = \left| \frac{1}{2}(\alpha s_k)^T \left[ 2c_i(x_k + \zeta\alpha s_k) \nabla c_i^2(x_k + \zeta\alpha s_k) + 2\nabla c_i(x_k + \zeta\alpha s_k) \nabla c_i(x_k + \zeta\alpha s_k)^T \right] (\alpha s_k) \right| \\
&\quad = \left| \alpha^2 s_k^T \left[ c_i(x_k + \zeta\alpha s_k) \nabla c_i^2(x_k + \zeta\alpha s_k) + \nabla c_i(x_k + \zeta\alpha s_k) \nabla c_i(x_k + \zeta\alpha s_k)^T \right] s_k \right| \\
&\quad \le \frac{1}{m} \tau_1 \alpha^2 \|s_k\|^2,
\end{aligned} \tag{3.3}$$

where the last inequality can be done by Assumption A1 and $\zeta \in [0,1]$. Furthermore, from (2.1) we immediately obtain $c_i(x_k) + \nabla c_i(x_k)^T s_k = 0$, that is, $-2\alpha c_i^2(x_k) - 2\alpha c_i(x_k)\nabla c_i(x_k)^T s_k = 0$. With $|S_2| \le m$, thereby,

$$\begin{aligned}
&|\theta_k(x_k + \alpha s_k) - (1-2\alpha)\theta_k(x_k)| \\
&\quad = \left|\sum_{i\in S_2}\left(c_i^2(x_k + \alpha s_k) - (1-2\alpha)c_i^2(x_k)\right)\right| \\
&\quad \le \sum_{i\in S_2}\left|c_i^2(x_k + \alpha s_k) - (1-2\alpha)c_i^2(x_k)\right| \\
&\quad = \sum_{i\in S_2}\left|c_i^2(x_k + \alpha s_k) - c_i^2(x_k) - 2\alpha c_i(x_k)\nabla c_i(x_k)^T s_k\right| \\
&\quad \le m \cdot \frac{1}{m}\tau_1\alpha^2\|s_k\|^2 \\
&\quad \le \tau_1\alpha^2\|s_k\|^2,
\end{aligned} \tag{3.4}$$

then the first inequality consequently holds.

According to the Taylor expansion of $\sum_{i\in S_1}(c_i^2(x_k + \alpha s_k))$ (i.e., $m_k(x_k + \alpha s_k)$), we then have

$$\left|\sum_{i\in S_1}\left(c_i^2(x_k + \alpha s_k)\right) - \sum_{i\in S_1}\left(c_i^2(x_k)\right) - \alpha g_k^T s_k\right| = \left|\frac{1}{2}\alpha^2(s_k)^T\nabla^2\sum_{i\in S_1}\left(c_i^2(x_k + \varrho\alpha s_k)\right)s_k\right| \le \tau_2\alpha^2\|s_k\|^2, \tag{3.5}$$

where the last inequality follows from Assumption A1 and $\varrho \in [0,1]$. That is to say,

$$\left|m_k(x_k + \alpha s_k) - m_k(x_k) - \alpha g_k^T s_k\right| \le \tau_2\alpha^2\|s_k\|^2, \tag{3.6}$$

which is just (3.2). □

**Lemma 3.2.** *Let $\{x_{k_i}\}$ be a subsequence of iterates for which* (2.3) *holds and has the same $S_1$ and $S_2$. Then there exists some $\widehat{\alpha} \in (0,1]$ such that*

$$m_{k_i}(x_{k_i} + \widehat{\alpha}s_{k_i}) \le m_{k_i}(x_{k_i}) + \widehat{\alpha}\tau_3 g_{k_i}^T s_{k_i}. \tag{3.7}$$

*Proof.* Because $\{x_{k_i}\}$ have the same $S_1$ and $S_2$, it follows that $m_{k_i}(x)$ are fixed and by (2.3) $d_{k_i}$ is a decent direction. Hence there exists some $\widehat{\alpha} \in (0,1]$ satisfying (3.7). □

**Theorem 3.3.** *Suppose that $\{x_k\}$ is an infinite sequence generated by Algorithm 2.1 and $|\mathcal{A}| < \infty$, one has*

$$\lim_{k\to\infty}\left\|c_{S_2^k}^k\right\| + \|s_k\| = 0, \tag{3.8}$$

*namely, every limit point is the* $\epsilon$ *solution to (1.1) or a local infeasible point. If the gradients of* $c_i(x_k)$ *are linear independent for all* $k$ *and* $i = 1, 2, \ldots, m$*, then the solution to SNE is obtained.*

*Proof.* From $|\mathcal{A}| < \infty$, we know the filter updates in a finite number, then there exists $K \in \mathbb{N}$, for $k > K$ the filter does not update. As h-type iteration and restoration algorithm all need the updating of the filter, so for $k > K$ our algorithm only follows the f-type iterations. We then have that for all $k > K$ both conditions (2.3) and (2.4) are satisfied for $x_{k+1} = x_k + \alpha_k s_k$ and $m_k(x) = m_K(x)$.

Then by (2.4) we get $m_k(x_{k+1}) \le \max\{m_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} m_{k-r}(x_{k-r})\} + \tau_3 \alpha_k g_k^T s_k$. We first show that for all $k \ge K+1$, it holds

$$m_k(x_k) < \overline{m}(x_K) + \lambda\tau_3 \sum_{r=K}^{k-2} \alpha_r g_r^T s_r + \tau_3 \alpha_{k-1} g_{k-1}^T s_{k-1} < \overline{m}(x_K) + \lambda\tau_3 \sum_{r=K}^{k-1} \alpha_r g_r^T s_r, \tag{3.9}$$

where $\overline{m}(x_K) = \max\{m_K(x_K), \sum_{r=0}^{m(k)-1} \lambda_{Kr} m_{K-r}(x_{K-r})\}$. We prove (3.9) by induction.

If $k = K+1$, we have $m_{K+1}(x_{K+1}) < \overline{m}(x_K) + \tau_3 \alpha_k g_k^T s_k < \overline{m}(x_K) + \lambda\tau_3 \alpha_k g_k^T s_k$. Suppose that the claim is true for $K+1, K+2, \ldots, k$, then we consider two cases.

*Case* 1. If $\max\{m_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} m_{k-r}(x_{k-r})\} = m_k(x_k)$, it is clear that

$$\begin{aligned} m_{k+1}(x_{k+1}) &< m_k(x_k) + \tau_3 \alpha_k g_k^T s_k < \overline{m}(x_K) + \lambda\tau_3 \sum_{r=K}^{k-1} \alpha_r g_r^T s_r + \tau_3 \alpha_k g_k^T s_k \\ &\le \overline{m}(x_K) + \lambda\tau_3 \sum_{r=K}^{k} \alpha_r g_r^T s_r. \end{aligned} \tag{3.10}$$

*Case* 2. If $\max\{m_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} m_{k-r}(x_{k-r})\} = \sum_{r=0}^{m(k)-1} \lambda_{kr} m_{k-r}(x_{k-r})$, let $u = m(k)-1$. By the fact that $\sum_{t=0}^{u} \lambda_{kt} = 1$, $\lambda \le \lambda_{kt} < 1$, we have

$$\begin{aligned} m_{k+1}(x_{k+1}) &< \sum_{t=0}^{u} \lambda_{kt} m_{k-t}(x_{k-t}) + \tau_3 \alpha_k g_k^T s_k \\ &< \sum_{t=0}^{u} \lambda_{kt} \left( \overline{m}(x_K) + \lambda\tau_3 \sum_{r=K}^{k-t-2} \alpha_r g_r^T s_r + \tau_3 \alpha_{k-t-1} g_{k-t-1}^T s_{k-t-1} \right) + \tau_3 \alpha_k g_k^T s_k \\ &= \lambda_{k0} \left( \overline{m}(x_K) + \lambda\tau_3 \sum_{r=K}^{k-u-2} \alpha_r g_r^T s_r + \lambda\tau_3 \sum_{r=k-u-1}^{k-2} \alpha_r g_r^T s_r + \tau_3 \alpha_{k-1} g_{k-1}^T s_{k-1} \right) \end{aligned}$$

$$+\lambda_{k1}\left(\overline{m}(x_K)+\lambda\tau_3\sum_{r=K}^{k-u-2}\alpha_r g_r^T s_r+\lambda\tau_3\sum_{r=k-u-1}^{k-3}\alpha_r g_r^T s_r+\tau_3\alpha_{k-2}g_{k-2}^T s_{k-2}\right)$$

$$+\cdots+\lambda_{ku}\left(\overline{m}(x_K)+\lambda\tau_3\sum_{r=K}^{k-u-2}\alpha_r g_r^T s_r+\tau_3\alpha_{k-u-1}g_{k-u-1}^T s_{k-u-1}\right)+\tau_3\alpha_k g_k^T s_k$$

$$<\overline{m}(x_K)+\lambda\tau_3\sum_{r=K}^{k-u-2}\alpha_r g_r^T s_r+\lambda\tau_3\sum_{r=k-u-1}^{k-1}\alpha_r g_r^T s_r+\tau_3\alpha_k g_k^T s_k$$

$$=\overline{m}(x_K)+\lambda\tau_3\sum_{r=K}^{k-1}\alpha_r g_r^T s_r+\tau_3\alpha_k g_k^T s_k<\overline{m}(x_K)+\lambda\tau_3\sum_{r=K}^{k}\alpha_r g_r^T s_r. \tag{3.11}$$

Moreover, since $m_k(x_k)$ is bounded below as $k \to \infty$, we get $\sum_{r=K}^{k}\alpha_r g_r^T s_r < \infty$, that is, $\lim_{k\to\infty}\alpha_r g_r^T s_r = 0$. By Lemma 3.2, there exists a $\widehat{\alpha} \in (0,1]$ such that $\alpha_k \geq \widehat{\alpha}$. Then together with $g_k^T s_k < -\xi s_k^T B_k s_k$ and Assumption A1, we have $\lim_{k\to\infty}\|s_k\| = 0$. From $-\alpha_{k,l}g_k^T s_k > [\theta_k(x_k)]^{s_\theta}$ it is easy to obtain that $\lim_{k\to\infty}\theta_k(x_k) = 0$. This completes the proof. □

**Lemma 3.4.** *Under Assumptions A1 and A2, if $g_k^T s_k \leq -\varepsilon_0$ for a positive constant $\varepsilon_0$ independent of $k$($\in$ a subsequence) and for all $\alpha \in (0,1]$ and $\alpha \geq \alpha_{k,l}^{\min}$ with $(\theta_k(x_k), m_k(x_k)) \notin \mathcal{F}_k$, then there exists $\gamma_1, \gamma_2 > 0$ so that $(\theta_k(x_k+\alpha s_k), m_k(x_k+\alpha s_k)) \notin \mathcal{F}_k$ for all $k$($\in$ a subsequence) and $\alpha \leq \min\{\gamma_1, \gamma_2\theta_k(x_k)\}$.*

*Proof.* Choose $\gamma_1 = \varepsilon_0/\tau_2\gamma_s^2$, then $\alpha \leq \gamma_1$ implies that $-\alpha\varepsilon_0 + \tau_2\alpha^2\gamma_s^2 \leq 0$. So we note from (3.2) that

$$\begin{aligned} m_k(x_k+\alpha s_k) &\leq m_k(x_k)+\alpha g_k^T s_k+\tau_2\alpha^2\|s_k\|^2 \\ &\leq m_k(x_k)-\alpha\varepsilon_0+\tau_2\alpha^2\gamma_s^2 \\ &\leq m_k(x_k). \end{aligned} \tag{3.12}$$

Let $\gamma_2 = 2/\tau_1\gamma_s^2$, then $\alpha \leq \gamma_2\theta_k(x_k)$ implies that $-2\alpha\theta_k(x_k) + \tau_1\alpha^2\gamma_s^2 \leq 0$. So from (3.1), we obtain

$$\begin{aligned} \theta_k(x_k+\alpha s_k) &\leq \theta_k(x_k)-2\alpha\theta_k(x_k)+\tau_2\alpha^2\|s_k\|^2 \\ &\leq \theta_k(x_k)-2\alpha\theta_k(x_k)+\tau_1\alpha^2\gamma_s^2 \\ &\leq \theta_k(x_k). \end{aligned} \tag{3.13}$$

We further point a fact according to the definition of filter. If $(\overline{\theta}, \overline{m}) \notin \mathcal{F}_k$ and $\theta \leq \overline{\theta}$, $m \leq \overline{m}$, we obtain $(\theta, m) \notin \mathcal{F}_k$. Thus from $(\theta_k(x_k), m_k(x_k)) \notin \mathcal{F}_k$, $m_k(x_k+\alpha s_k) \leq m_k(x_k)$, and $\theta_k(x_k+\alpha s_k) \leq \theta_k(x_k)$, we have $(\theta_k(x_k+\alpha s_k), m_k(x_k+\alpha s_k)) \notin \mathcal{F}_k$. □

**Lemma 3.5.** *If $g_k^T s_k \le -\varepsilon_0$ for a positive constant $\varepsilon_0$ independent of $k(\in a\ subsequence)$, then there exists a constant $\overline{\alpha} > 0$, for all $k(\in a\ subsequence)$ and $\alpha \le \overline{\alpha}$ such that*

$$m_k(x_k + \alpha s_k) - \max\left\{ m_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} m_{k-r}(x_{k-r}) \right\} \le \tau_3 \alpha g_k^T s_k. \tag{3.14}$$

*Proof.* Let $\overline{\alpha} = (1-\tau_3)\varepsilon_0/\tau_2\gamma_s^2$. In view of (3.2), $\|s_k\| \le \gamma_s$ and $\alpha \le \overline{\alpha}$, we know

$$\begin{aligned} & m_k(x_k + \alpha s_k) - \max\left\{ m_k(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} m_{k-r}(x_{k-r}) \right\} - \alpha g_k^T s_k \\ & \le m_k(x_k + \alpha s_k) - m_k(x_k) - \alpha g_k^T s_k \\ & \le \tau_2 \alpha^2 \|s_k\|^2 \\ & \le \tau_2 \alpha \overline{\alpha} \gamma_s^2 = (1-\tau_3)\alpha\varepsilon_0 \le -(1-\tau_3)\alpha g_k^T s_k, \end{aligned} \tag{3.15}$$

which shows that the assertion of the lemma follows. □

**Theorem 3.6.** *Suppose that $\{x_k\}$ is an infinite sequence generated by Algorithm 2.1 and $|\mathcal{A}| = \infty$. Then there exists at least one accumulation which is the $\epsilon$ solution to (1.1) or a local infeasible point. Namely, one has*

$$\lim_{k\to\infty} \inf \left[ \left\| c^k_{S_2^k} \right\| + \|s_k\| \right] = 0. \tag{3.16}$$

*If the gradients of $c_i(x_k)$ are linear independent for all $k$ and $i = 1, 2, \ldots, m$, then the solution to* (1.1) *is obtained.*

*Proof.* We prove that $\lim_{k\to\infty, k\in\mathcal{A}} \theta_k(x_k) = 0$ first.

Suppose by contradiction that there exits an infinite subsequence $\{k_i\}$ of $\mathcal{A}$ such that $\theta_{k_i}(x_{k_i}) \ge \varepsilon$ for some $\varepsilon > 0$. At each iteration $k_i$, $(\theta_{k_i}(x_{k_i}), m_{k_i}(x_{k_i}))$ is added to the filter which means that no other $(\theta, m)$ can be added to the filter at a later stage within the area:

$$\left[\overline{\theta}(x_{k_i}) - \gamma_\theta \varepsilon, \overline{\theta}(x_{k_i})\right] \times \left[\overline{m}(x_{k_i}) - \gamma_m \varepsilon, \overline{m}(x_{k_i})\right], \tag{3.17}$$

and the area of the each of these squares is at least $\gamma_\theta \gamma_m \varepsilon^2$.

By Assumption A1 we have $\sum_{i=1}^n c_i^2(x_k) \le M^{\max}$. Since $0 \le m_k(x_k) \le m_k(x_k) + \theta_k(x_k) = \sum_{i=1}^n c_i^2(x_k)$ and $0 \le \theta_k(x_k) \le m_k(x_k) + \theta_k(x_k) = \sum_{i=1}^n c_i^2(x_k)$, then $(\theta, m)$ associated with the filter are restricted to

$$\mathcal{B} = [0, M^{\max}] \times [0, M^{\max}]. \tag{3.18}$$

Thereby $\mathcal{B}$ is completely covered by at most a finite number of such areas in contraction to the infinite subsequence $\{k_i\}$ satisfying $\theta_{k_i}(x_{k_i}) \ge \varepsilon$. Therefore, $\lim_{k\to\infty, k\in\mathcal{A}} \theta_k(x_k) = 0$.

By Assumption A1 and $|\mathcal{A}| = \infty$, there exits an accumulation point $\overline{x}$, that is, $\lim_{i\to\infty} x_{k_i} = \overline{x}$, $k_i \in \mathcal{A}$. It follows from $\lim_{k\to\infty, k\in\mathcal{A}} \theta_k(x_k) = 0$, that

$$\lim_{i\to\infty} \theta_{k_i}(x_{k_i}) = 0, \tag{3.19}$$

which implies $\lim_{i\to\infty}\|c^{k_i}_{S_2^{k_i}}\| = 0$. If $\lim_{i\to\infty}\|s_{k_i}\| = 0$, then (3.16) is true. Otherwise, there exists a subsequence $\{x_{k_{i_j}}\}$ of $\{x_{k_i}\}$ and a constant $\varepsilon_1 > 0$ so that for all $k_{i_j}$,

$$\left\|s_{k_{i_j}}\right\| \geq \varepsilon_1. \tag{3.20}$$

The choice of $\{x_{k_{i_j}}\}$ implies

$$k_{i_j} \in \mathcal{A} \quad \text{for all } k_{i_j}. \tag{3.21}$$

According to $\|s_{k_{i_j}}\| \geq \varepsilon_1$, Assumption A1 as well as $\xi \in (0,1)$, we have

$$\begin{aligned} g_{k_{i_j}}^T s_{k_{i_j}} + \xi s_{k_{i_j}}^T B_{k_{i_j}} s_{k_{i_j}} &= (\xi - 1) s_{k_{i_j}}^T B_{k_{i_j}} s_{k_{i_j}} - \left(\lambda_{k_{i_j}}^+\right)^T c^{k_{i_j}}_{S_2^{k_{i_j}}} \\ &\leq (\xi - 1) a \left\|s_{k_{i_j}}\right\|^2 + c_1 \left\|c^{k_{i_j}}_{S_2^{k_{i_j}}}\right\| \\ &\leq (\xi - 1) a \varepsilon_1^2 + c_1 \left\|c^{k_{i_j}}_{S_2^{k_{i_j}}}\right\|. \end{aligned} \tag{3.22}$$

Since $\xi - 1 < 0$ and $\|c^{k_{i_j}}_{S_2^{k_{i_j}}}\| \to 0$ as $j \to \infty$, we obtain

$$g_{k_{i_j}}^T s_{k_{i_j}} \leq -\xi s_{k_{i_j}}^T B_{k_{i_j}} s_{k_{i_j}}, \tag{3.23}$$

for sufficiently large $j$. Similarly, we have

$$\begin{aligned} \alpha g_{k_{i_j}}^T s_{k_{i_j}} + [\theta_k(x_k)]^{s_\theta} &\leq -\alpha s_{k_{i_j}}^T B_{k_{i_j}} s_{k_{i_j}} + c_1 \left\|c^{k_{i_j}}_{k_{i_j 2}}\right\| + [\theta_k(x_k)]^{s_\theta} \\ &\leq -\alpha a \varepsilon_1^2 + c_1 \left\|c^{k_{i_j}}_{k_{i_j 2}}\right\| + [\theta_k(x_k)]^{s_\theta}, \end{aligned} \tag{3.24}$$

and thus

$$-\alpha g_{k_{i_j}}^T s_{k_{i_j}} \geq [\theta_k(x_k)]^{s_\theta}, \tag{3.25}$$

**Table 1:** Numerical results of Example 4.1.

| Starting point | NIT | NOF | NOG |
|---|---|---|---|
| $(1,0)$ | 2 | 5 | 7 |
| $(1,2)$ | 6 | 13 | 12 |

**Table 2:** Numerical results of Example 4.2.

| Starting point | NIT | NOF | NOG |
|---|---|---|---|
| $(0,0,0)$ | 10 | 21 | 22 |
| $(1.5,1.5,1.5)$ | 7 | 15 | 15 |

for sufficiently large $j$. This means the condition (2.3) is satisfied for sufficiently large $j$. Therefore, the reason for accepting $x_{k+1}$ must been that $x_{k+1}$ satisfies nonmonotone Armijo condition (2.4). In fact let $\varepsilon_0 = \xi a \varepsilon_1^2$, then $g_{k_{i_j}}^T s_{k_{i_j}} \leq -\xi s_{k_{i_j}}^T B_{k_{i_j}} s_{k_{i_j}} \leq -\xi a \varepsilon_1^2 = -\varepsilon_0$; by Lemma 3.5 we obtain nonmonotone Armijo condition (2.4) is satisfied. Consequently, the filter is not augmented in iteration $k_{i_j}$ which is a contraction to (3.21). The whole proof is completed. □

## 4. Numerical Experiments

In this section, we test our algorithm on some typical test problems. In the whole process, the program is coded in MATLAB and we assume the error tolerance $\epsilon$ in this paper is always $1.0e-5$. The selected parameter values are $\gamma_\theta = 0.1$, $\gamma_m = 0.1$, $s_\theta = 0.9$, $\rho_1 = 0.25$, $\rho_2 = 0.75$, and $M = 3$. In the following tables, the notations NIT, NOF, and NOG mean the number of iterates, number of functions, and number of gradients, respectively.

*Example 4.1.* Find a solution of the nonlinear equations system as follows:

$$\begin{pmatrix} x + 3y^2 \\ (x - 1.0)y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.1}$$

The only solution of Example 4.1 is $(x^*, y^*) = (0,0)$. Define the line $\Gamma = \{(1,y) : y \in \mathbb{R}\}$. If the starting point $(x_0, y_0) \in \Gamma$, the Newton method [24] are confined to $\Gamma$. We choose two starting points which belong to $\Gamma$ in the experiments and then the $(x^*, y^*)$ is obtained. Table 1 shows the results.

*Example 4.2.* Consider the system of nonlinear equations:

$$\begin{aligned} c_1(x) &= x_1^3 - x_2^3 + x_3^3 - 1, \\ c_2(x) &= x_1^2 + x_2^2 - x_3^2 - 1, \\ c_3(x) &= x_1 + x_2 + x_3 - 3. \end{aligned} \tag{4.2}$$

The solution to Example 4.2 is $x^* = (1,1,1)^T$. The numerical results of Example 4.2 are given in Table 2.

**Table 3:** Numerical results of Example 4.3.

| Starting point | NIT | NOF | NOG | NIT [22] |
|---|---|---|---|---|
| (3, 1) | 6 | 10 | 8 | 17 |
| (30, 10) | 7 | 14 | 13 | 7 |
| (300, 100) | 10 | 17 | 16 | 15 |

**Table 4:** Numerical results of Example 4.4.

| Starting point | (0.5, 0.5) | (−0.5, 0.5) | (0.5, −0.5) |
|---|---|---|---|
| NIT | 5 | 5 | 6 |
| NOF | 8 | 7 | 9 |
| NOG | 7 | 6 | 9 |
| Solution | (1, 1) | (−1, 1) | (1, −1) |
| NIT [22] | 5 | 6 | 9 |

*Example 4.3.* Find a solution of the nonlinear equations system:

$$\begin{pmatrix} x \\ \dfrac{10x}{(x+0.1)} + 2y^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.3}$$

The unique solution is $(x^*, y^*) = (0, 0)$. It has been proved in [2] that, under initial point $(x_0, y_0) = (3, 1)$, the iterates converge to the point $z = (1.8016, 0.0000)$, which is not a stationary point. Utilizing our algorithm, a sequence of points converging to $(x^*, y^*)$ is obtained. The detailed numerical results for Example 4.3 are listed in Table 3.

*Example 4.4.* Consider the following system of nonlinear equations:

$$\begin{aligned} c_1(x) &= x_1^2 + x_1x_2 + 2x_2^2 - x_1 - x_2 - 2, \\ c_2(x) &= 2x_1^2 + x_1x_2 + 3x_2^2 - x_1 - x_2 - 4. \end{aligned} \tag{4.4}$$

There are three solutions of above example, $(1, 1)^T$, $(-1, 1)^T$, and $(1, -1)^T$. The numerical results of Example 4.4 are given in Table 4.

*Example 4.5.* Consider the system of nonlinear equations:

$$c_i(x) = -(N+1) + 2x_i + \sum_{j=1, j\neq i}^{N} x_j, \quad i = 1, 2, \ldots, N-1, \tag{4.5}$$

$$c_N(x) = -1 + \prod_{j=1}^{N} x_j, \tag{4.6}$$

with the initial point $x_i^{(0)} = 0.5$, $i = 1, 2, \ldots, N$. The solution to Example 4.5 is $x^* = (1, 1, \ldots, 1)^T$. The numerical results of Example 4.5 are given in Table 5.

**Table 5:** Numerical results of Example 4.5.

| | $N = 10$ | $N = 20$ | $N = 40$ | $N = 60$ | $N = 120$ |
|---|---|---|---|---|---|
| NIT | 7 | 10 | 19 | 28 | 52 |
| NOF | 14 | 22 | 26 | 39 | 77 |
| NOG | 13 | 21 | 23 | 34 | 68 |
| NIT [22] | 8 | 17 | 22 | 41 | Fail |

Refer to these above problems, running the Algorithm 2.1 with different starting points yields the results in the corresponding tables, which, summarized, show that our proposed algorithm is practical and effective. From the computation efficiency, we should point out our algorithm is competitive with the method in [22]. The results in Table 5 in fact show that our method also succeeds well to solve the cases when more equations are active.

Constrained optimization approaches attacking the system of nonlinear equations are exceedingly interesting and are further developed by using the nonmonotone line search filter strategy in this paper. Moreover, the local property of the algorithm is a further topic of interest.

## Acknowledgment

The research is supported by the National Natural Science Foundation of China (no. 11126060) and Science & Technology Program of Shanghai Maritime University (no. 20120060).

## References

[1] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, NY, USA, 1999.

[2] M. J. D. Powell, "A hybrid method for nonlinear equations," in *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz, Ed., pp. 87–114, Gordon and Breach, London, UK, 1970.

[3] R. Fletcher and S. Leyffer, "Nonlinear programming without a penalty function," *Mathematical Programming*, vol. 91, no. 2, pp. 239–269, 2002.

[4] R. Fletcher, S. Leyffer, and P. L. Toint, "On the global convergence of a filter-SQP algorithm," *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 44–59, 2002.

[5] R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint, and A. Wächter, "Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming," *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 635–659, 2002.

[6] S. Ulbrich, "On the superlinear local convergence of a filter-SQP method," *Mathematical Programming*, vol. 100, no. 1, pp. 217–245, 2004.

[7] C. Audet and J. E. Dennis Jr., "A pattern search filter method for nonlinear programming without derivatives," *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 980–1010, 2004.

[8] C. M. Chin and R. Fletcher, "On the global convergence of an SLP-filter algorithm that takes EQP steps," *Mathematical Programming*, vol. 96, no. 1, pp. 161–177, 2003.

[9] M. Ulbrich, S. Ulbrich, and L. N. Vicente, "A globally convergent primal-dual interior filter method for nonconvex nonlinear programming," *Mathematical Programming*, vol. 100, no. 2, pp. 379–410, 2003.

[10] R. Fletcher and S. Leyffer, "A bundle filter method for nonsmooth nonlinear optimization," Tech. Rep. NA/195, Department of Mathematics, University of Dundee, Scotland, UK, 1999.

[11] E. Karas, A. Ribeiro, C. Sagastizábal, and M. Solodov, "A bundle-filter method for nonsmooth convex constrained optimization," *Mathematical Programming*, vol. 116, no. 1-2, pp. 297–320, 2009.

[12] N. I. M. Gould, S. Leyffer, and P. L. Toint, "A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 17–38, 2004.

[13] N. I. M. Gould, C. Sainvitu, and P. L. Toint, “A filter-trust-region method for unconstrained optimization,” *SIAM Journal on Optimization*, vol. 16, no. 2, pp. 341–357, 2005.
[14] A. Wächter and L. T. Biegler, “Line search filter methods for nonlinear programming: motivation and global convergence,” *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 1–31, 2005.
[15] A. Wächter and L. T. Biegler, “Line search filter methods for nonlinear programming: local convergence,” *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 32–48, 2005.
[16] L. Grippo, F. Lampariello, and S. Lucidi, “A nonmonotone line search technique for Newton’s method,” *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.
[17] C. Gu and D. T. Zhu, “A non-monotone line search multidimensional filter-SQP method for general nonlinear programming,” *Numerical Algorithms*, vol. 56, no. 4, pp. 537–559, 2011.
[18] Z. S. Yu and D. G. Pu, “A new nonmonotone line search technique for unconstrained optimization,” *Journal of Computational and Applied Mathematics*, vol. 219, no. 1, pp. 134–144, 2008.
[19] R. Fletcher and S. Leyffer, “Filter-type algorithms for solving systems of algebraic equations and inequalities,” Dundee Numerical Analysis Report NA/204, 2001.
[20] P.-Y. Nie, “A null space method for solving system of equations,” *Applied Mathematics and Computation*, vol. 149, no. 1, pp. 215–226, 2004.
[21] P.-Y. Nie, “CDT like approaches for the system of nonlinear equations,” *Applied Mathematics and Computation*, vol. 172, no. 2, pp. 892–902, 2006.
[22] P.-Y. Nie, “An SQP approach with line search for a system of nonlinear equations,” *Mathematical and Computer Modelling*, vol. 43, no. 3-4, pp. 368–373, 2006.
[23] P.-Y. Nie, M.-Y. Lai, S.-J. Zhu, and P.-A. Zhang, “A line search filter approach for the system of nonlinear equations,” *Computers & Mathematics with Applications*, vol. 55, no. 9, pp. 2134–2141, 2008.
[24] R. H. Byrd, M. Marazzi, and J. Nocedal, “On the convergence of Newton iterations to non-stationary points,” *Mathematical Programming*, vol. 99, no. 1, pp. 127–148, 2004.