Line search filter inexact secant methods for nonlinear equality constrained optimization*

Zhujun Wang^{a †} Li Cai^{b ‡} Detong Zhu^{c §}

^aCollege of Science, Hunan Institute of Engineering, Xiangtan 411104, China

^{b,c}Department of Mathematics, Shanghai Normal University, Shanghai 200234, China

Abstract

We present inexact secant methods in association with line search filter technique for solving nonlinear equality constrained optimization. For large-scale applications, it is expensive to get an exact search direction, and hence we use an inexact method that finds an approximate solution satisfying some appropriate conditions. The global convergence of the proposed algorithm is established by using line search filter technique. The second-order correction step is used to overcome the Maratos effect, while the line search filter inexact secant methods have superlinear local convergence rate. Finally, the results of numerical experiments indicate that the proposed methods are efficient for the given test problems.

Keywords: constrained optimization; filter method; inexact method; secant method; Maratos effect.

AMS Classification: 49M37, 90C30, 65K05.

1 Introduction

We consider the problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to } c(x) = 0, \quad (1.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are twice continuously differentiable.

Fletcher and Leyffer in [6] proposed a class of filter methods, which do not require any penalty parameter and have promising numerical results. Consequently, filter technique has employed to many approaches, for instance, SLP methods [4], SQP methods [5, 12],

*The authors gratefully acknowledge the partial supports of the National Science Foundation Grant (11371253) of China and the Science Foundation Grant (11C0336) of Provincial Education Department of Hunan.

[†]E-mail address: wangzj1998@aliyun.com

[‡]E-mail address: caili0039@163.com

[§]E-mail address: dtzhu@shnu.edu.cn

interior point approaches [13], bundle techniques [6] and so on. In [10, 11], Wächter and Biegler presented a line search filter method for nonlinear equality constrained programming and discussed the global convergence properties and superlinear local convergence rate.

Secant methods (two-step algorithms) as defined in [7] are one of the most successful methods for solving problem (1.1) and have a main advantage which rests in the use of an orthogonal projection operator which is continuous. The basic idea about the secant algorithms can be summarized as follows: at each iteration, the total step d_k is considered as the sum of a horizontal step u_k and a vertical step v_k . The general form of the secant algorithms follows, in each iteration

$$\lambda_{k+1} = U(x_k, \lambda_k, W_k), \quad (1.2a)$$

$$W_k \hat{u}_k = -\nabla_x \mathcal{L}(x_k, \lambda_{k+1}), \quad (1.2b)$$

$$u_k = P_k \hat{u}_k, \quad (1.2c)$$

$$v_k = -\nabla c_k^\dagger c_k, \quad (1.2d)$$

$$y_k = \nabla_x \mathcal{L}(x_k + u_k, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1}), \quad (1.2e)$$

$$W_{k+1} = B(u_k, y_k, W_k), \quad (1.2f)$$

$$x_{k+1} = x_k + u_k + v_k, \quad (1.2g)$$

where $\nabla c_k^\dagger$ is the pseudo-inverse of ∇c_k^T , $\mathcal{L}$ is the Lagrangian function defined for $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$ by $\mathcal{L}(x, \lambda) = f(x) + \lambda^T c(x)$, and B is a secant update formula generating matrices W_k approximating the Hessian of the Lagrangian at each x_k . The projection onto the null space of $\nabla c(x)^T$ can be either the orthogonal projection $P(x)$ given by

$$P(x) = I - \nabla c(x) [\nabla c(x)^T \nabla c(x)]^{-1} \nabla c(x)^T \quad (1.3a)$$

or the oblique projection defined by

$$P(x) = I - W^{-1} \nabla c(x) [\nabla c(x)^T W^{-1} \nabla c(x)]^{-1} \nabla c(x)^T. \quad (1.3b)$$

The multiplier updates $U(x_k, \lambda_k, W_k)$ in (1.2a) can be chosen from one of the following updates:

$$\lambda_{k+1}^P = -(\nabla c_k^T \nabla c_k)^{-1} \nabla c_k^T \nabla f_k, \quad (1.4a)$$

$$\lambda_{k+1}^S = -(\nabla c_k^T W_k^{-1} \nabla c_k)^{-1} \nabla c_k^T W_k^{-1} \nabla f_k, \quad (1.4b)$$

$$\lambda_{k+1}^N = (\nabla c_k^T W_k^{-1} \nabla c_k)^{-1} (c_k - \nabla c_k^T W_k^{-1} \nabla f_k), \quad (1.4c)$$

where $\lambda_{k+1}^P, \lambda_{k+1}^S, \lambda_{k+1}^N$ are Projection update, Null-space update and Newton update, respectively. The pseudo-inverse of ∇c_k^T , $\nabla c_k^\dagger$ will be either

$$\nabla c_k^\dagger = \nabla c_k (\nabla c_k^T \nabla c_k)^{-1} \quad (1.5a)$$

or

$$\nabla c_k^\dagger = W_k^{-1} \nabla c_k (\nabla c_k^T W_k^{-1} \nabla c_k)^{-1}. \quad (1.5b)$$

A drawback of many contemporary SQP algorithms is that they require explicit representations of exact derivative information and the solution of one or more linear systems during every iteration. For large-scale applications, it may be too expensive to get. In [2, 3], Byrd, Curtis and Nocedal present inexact methods and give some conditions that guarantee the global convergence of inexact steps.

Motivated by the idea and methods above, we propose an inexact secant method in association with line search filter technique, which has both global convergence and superlinear local convergence rate. The method is globalized by line search and filter approach. It has been noted by Fletcher and Leyffer [6] that the filter approach, similar to a penalty function approach, can suffer from Maratos effect, hence it is necessary to overcome Maratos effect by solving second order correction (SOC) step in our algorithm. In addition, it is shown that the iterative point generated by our algorithm is superlinear convergent.

The paper is organized as follows. In Section 2 the algorithm is developed. Section 3 analyzes the global convergence properties. We characterize the local superlinear convergence in Section 4. The results of numerical experience with the method are discussed in Section 5.

2 Algorithm

Throughout the paper, we denote the Euclidean vector or matrix norm by $\|\cdot\|$, l_1 norm by $\|\cdot\|_1$, $A(x) := \nabla c(x)$, $A(x)^\dagger := \nabla c(x)^\dagger$ and $g(x) := \nabla f(x)$.

We will compute the horizontal step u_k inexactly, which means that (1.2b) will be substituted by

$$W_k \hat{u}_k = -\nabla_x \mathcal{L}(x_k, \lambda_{k+1}) + r_k, \quad (2.1)$$

where $\|r_k\| \leq \eta_k \|c_k\|$ with $\eta_k \in [0, t]$ and $t \in (0, 1)$. We use the norm of c_k to control the residual because $\lim_{k \rightarrow \infty} \|c(x_k)\| = 0$, which will be shown in Theorem 3.1.

Because of adopting l_1 penalty function as a merit function, it is necessary to overcome Maratos effect by solving second order correction (SOC) step $\hat{d}_k$ in our algorithm. In this work, $\hat{d}_k$ can be either

$$\hat{d}_k = -A_k(A_k^T A_k)^{-1} c(x_k + d_k) \quad (2.2a)$$

or

$$\hat{d}_k = -W_k^{-1} A_k(A_k^T W_k^{-1} A_k)^{-1} c(x_k + d_k). \quad (2.2b)$$

Table 2.1 presents six related algorithms with different $P(x)$, λ_{k+1} , $A_k^\dagger$ and $\hat{d}_k$.

In order to obtain the next iteration, we use the line search filter approach to find a step size $\alpha_k \in (0, 1]$.

From the filter idea, we know that a nonlinear programming algorithm must deal with two different criteria, related to optimality and to feasibility, respectively. Optimality is measured by the objective function f ; feasibility is typically measured by penalization of constraint violation, for instance, by the function h , given by $h(x) = \|c(x)\|$, where $\|\cdot\|$ is the ordinary Euclidean norm.

We say that a trial point x_k is acceptable for the filter if and only if

$$h(x_k + \alpha_{k,l} d_k) \leq (1 - \gamma_h) h(x_k) \quad (2.3a)$$

or

$$f(x_k + \alpha_{k,l} d_k) \leq f(x_k) - \gamma_f h(x_k), \quad (2.3b)$$

where the constants $\gamma_h, \gamma_f \in (0, 1)$. When (2.3a)-(2.3b) hold, the trial point should satisfy the Armijo-type condition if the f -type switching condition holds

$$\alpha_{k,l} g_k^T d_k < 0 \quad \text{and} \quad (-\alpha_{k,l} g_k^T d_k)^{s_f} \alpha_{k,l}^{1-s_f} > \tau h(x_k)^{s_h}, \quad (2.4)$$

where the constants $s_f \geq 1$, $s_h > 1$ and $\tau > 0$. The Armijo-type condition is

$$f(x_k + \alpha_{k,l} d_k) \leq f(x_k) + \beta \alpha_{k,l} g_k^T d_k, \quad (2.5)$$

where $\beta \in (0, \frac{1}{2})$.

In this paper, we employ the definition of filter in [10]. The filter is defined as a set $\mathcal{F}_k \subseteq [0, \infty) \times \mathbb{R}$ containing all (h, f) -pairs that are prohibited in iteration k . If $(h(x_k + \alpha_{k,l} d_k), f(x_k + \alpha_{k,l} d_k)) \notin \mathcal{F}_k$, then the trial point is acceptable to the filter. We use the updating formula to augment the filter

$$\mathcal{F}_{k+1} = \mathcal{F}_k \cup \{(h, f) \in \mathbb{R}^2 : h \geq (1 - \gamma_h)h(x_k) \text{ and } f \geq f(x_k) - \gamma_f h(x_k)\}, \quad (2.6)$$

which can be seen in [6].

We now formally state the global inexact line search filter secant method for solving (1.1).

Algorithm A

1. Initialize. Choose starting point x_0 and λ_0 . Set constants $\beta \in (0, \frac{1}{2})$, $\gamma_h, \gamma_f \in (0, 1)$, $\tau, \nu > 0$, $s_h > 1$, $s_f > 2s_h$, $\gamma_\alpha \in (0, 1]$, $\varsigma \in (0, 1)$ and $\varepsilon > 0$. Choose an initial filter $\mathcal{F}_0 := \{(h, f) \in \mathbb{R}^2 : h \geq \nu\}$ and let $k := 0$.
2. If $\|\nabla_x l(x_k, \lambda_k)\|_1 + \|c_k\|_1 < \varepsilon$, then stop.
3. Evaluate f_k, g_k, c_k and A_k . Then compute the multiplier $\lambda_{k+1} = U(x_k, \lambda_k, W_k)$ from (1.4). Find an inexact step d_k from

$$W_k \hat{u}_k = -\nabla_x \mathcal{L}(x_k, \lambda_{k+1}) + r_k, \quad (2.7a)$$

$$u_k = P_k \hat{u}_k, \quad (2.7b)$$

$$v_k = -A_k^\dagger c_k, \quad (2.7c)$$

$$d_k = u_k + v_k, \quad (2.7d)$$

satisfying

$$\|r_k\| \leq \eta_k \|c_k\| \quad (2.8)$$

with $\eta_k \in [0, t]$ and $t \in (0, \frac{1-\gamma_h}{1+\gamma_h})$. In (2.7), P_k is given by (1.3) and $A_k^\dagger$ is given by (1.5). If (2.7a) is too ill-conditioned, go to Step 11.

4. Set $\alpha_{k,0} = 1$ and $l = 0$.

5. If $\alpha_{k,l} < \alpha_k^{\min}$ with

$$\alpha_k^{\min} := \gamma_\alpha \cdot \begin{cases} \min\{\gamma_h, \frac{\gamma_f h_k}{-g_k^T d_k}, \frac{\tau h_k^{s_h}}{(-g_k^T d_k)^{s_f}}\} & \text{if } g_k^T d_k < 0 \\ \gamma_h & \text{otherwise,} \end{cases} \quad (2.9)$$

go to Step 11. Otherwise, if $(h(x_k + \alpha_{k,l} d_k), f(x_k + \alpha_{k,l} d_k)) \in \mathcal{F}_k$, then go to Step 7.

6. If (2.4) holds, $\alpha_{k,l}$ is an f -step size. Then if (2.5) holds, set $x_{k+1} := x_k + \alpha_{k,l}d_k$ and go to Step 10. On the other hand, if (2.4) does not hold, then if (2.3) holds, set $x_{k+1} := x_k + \alpha_{k,l}d_k$ and go to Step 10.
7. If $l \neq 0$, go to Step 9. Otherwise, compute the second order correction step $\hat{d}_k$ by (2.2). Then if $(h(x_k + d_k + \hat{d}_k), f(x_k + d_k + \hat{d}_k)) \in \mathcal{F}_k$, go to Step 9.
8. If (2.4) holds with $x_k + d_k$ and $f(x_k + d_k + \hat{d}_k) \leq f_k + \beta g_k^T d_k$, holds, then set $x_{k+1} := x_k + d_k + \hat{d}_k$ and go to Step 10.
If (2.4) does not hold with $x_k + d_k$, then if $h(x_k + d_k + \hat{d}_k) \leq (1 - \gamma_h)h(x_k)$ or $f(x_k + d_k + \hat{d}_k) \leq f(x_k) - \gamma_f h(x_k)$ holds, set $x_{k+1} := x_k + d_k + \hat{d}_k$ and go to Step 10.
9. Set $\alpha_{k,l+1} = \varsigma \alpha_{k,l}$ and $l \leftarrow l + 1$. Go to Step 5.
10. Set $\alpha_k := \alpha_{k,l}$. If (2.4) does not hold for α_k , augment the filter by (2.6); otherwise set $\mathcal{F}_{k+1} := \mathcal{F}_k$. Go to Step 12.
11. Feasibility restoration phase. Find a new point x_{k+1} such that
 - (a) $h(x_{k+1})$ is smaller than $h(x_k)$;
 - (b) (2.3) holds for x_{k+1} ;
 - (c) $(h(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_k$.

Augment the filter by (2.6). Go to Step 12.

12. Compute $y_k = P_k[\nabla_x \mathcal{L}(x_k + u_k, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1})]$, and update W_k by $W_{k+1} = \text{BFGS}(u_k, y_k, W_k)$. Set $k := k + 1$ and go to Step 2.

Remark 1. Backtracking line search technique may cause $\lim_l \alpha_{k,l} = 0$. In the case that $h_k > 0$, it can be seen from (2.9) that $\alpha_k^{\min} > 0$. Hence, the algorithm switches to the feasibility restoration phase if $\alpha_{k,l} < \alpha_k^{\min}$ in step 5. In the case that $h_k = 0$, (2.9) implies $\alpha_k^{\min} = 0$. The Armijo condition (2.5) can be satisfied for a sufficiently small step size α_k .

Remark 2. The restoration phase algorithm converges to a stationary point for the constraint violation, assuming that a suitable method is used.

3 Global convergence

We denote $\mathcal{R}$ as the set of iterations in which the restoration phase is invoked in the algorithm. The set $\bar{\mathcal{R}}$ is defined as the set of iterations in which the restoration phase is invoked because the system (2.7) is too ill-conditioned. We denote the set of the iterations in which the filter has been augmented by $\mathcal{A}$.

The criticality measure is defined as

$$\chi(x_k) := \begin{cases} \|u_k\|_2 & k \notin \bar{\mathcal{R}} \\ \infty & \text{otherwise.} \end{cases} \quad (3.1)$$

We begin with the global convergence analysis of the proposed algorithm under the following assumptions.

Assumptions G.

- G1. f and c are bounded and twice continuously differentiable on $\mathbb{R}^n$.
- G2. There exists a constant $M_h > 0$ such that $k \notin \bar{\mathcal{R}}$ whenever $h_k \leq M_h$.
- G3. There exists a constant $\kappa > 0$ such that $u^T W_k u \geq \kappa \|u\|^2$ for $A_k^T u = 0$ and

$\{W_k\}$ are uniformly bounded.

G4. A_k have full row rank and their smallest singular values are bounded below by some positive constant.

G5. The sequences $\{A_k^\dagger\}$ are uniformly bounded.

These assumptions are fairly standard for a line search method and similar to those made in [10]. From Assumptions G1, G3 and G4, we can easily prove the next lemma.

Lemma 3.1. Suppose Assumptions G hold. Then u_k , v_k and λ_{k+1} are bounded.

Next, we prove that the search direction is a direction of sufficient descent for f at points that are sufficiently close to feasible and nonoptimal.

Lemma 3.2. Suppose Assumptions G hold. If there exists a subsequence of iterates $\{x_{k_i}\}$ such that $\chi(x_{k_i}) \geq \epsilon$ with $\epsilon > 0$, then there exist constants $\epsilon_1, \epsilon_2 > 0$ satisfying

$$h(x_{k_i}) \leq \epsilon_1 \Rightarrow g_{k_i}^T d_{k_i} \leq -\epsilon_2 \quad (3.2)$$

for all i .

Proof. If there exists a subsequence of iterates $\{x_{k_i}\}$ such that $\chi(x_{k_i}) \geq \epsilon$ with $\epsilon > 0$, then by G2, we have $k_i \notin \bar{\mathcal{R}}$. From (2.7c) and Assumption G4, G5, we can obtain that

$$v_k = O(\|c_k\|). \quad (3.3)$$

For Algorithm 1 and 4, we have that

$$\begin{aligned} u_k &= \hat{u}_k \\ &= -W_k^{-1}g_k + W_k^{-1}A_k\lambda_{k+1} + W_k^{-1}r_k \\ &= -W_k^{-1}g_k + W_k^{-1}A_k(A_k^T W_k^{-1}A_k)^{-1}A_k^T W_k^{-1}g_k + W_k^{-1}r_k. \end{aligned}$$

Hence

$$\begin{aligned} u_k^T W_k u_k &= u_k^T [-g_k + A_k(A_k^T W_k^{-1}A_k)^{-1}A_k^T W_k^{-1}g_k + r_k] \\ &= -u_k^T g_k + g_k^T W_k^{-1}A_k(A_k^T W_k^{-1}A_k)^{-1}A_k^T W_k^{-1}r_k + u_k^T r_k. \end{aligned}$$

So

$$g_k^T u_k = -u_k^T W_k u_k + g_k^T W_k^{-1}A_k(A_k^T W_k^{-1}A_k)^{-1}A_k^T W_k^{-1}r_k + u_k^T r_k. \quad (3.4)$$

By (2.8), (3.3), (3.4), Assumption G3 and $\chi(x_{k_i}) \geq \epsilon$, we have that for $k_i \notin \bar{\mathcal{R}}$

$$\begin{aligned} g_{k_i}^T d_{k_i} &= g_{k_i}^T u_{k_i} + g_{k_i}^T v_{k_i} \\ &= -u_{k_i}^T W_{k_i} u_{k_i} + g_{k_i}^T W_{k_i}^{-1}A_{k_i}(A_{k_i}^T W_{k_i}^{-1}A_{k_i})^{-1}A_{k_i}^T W_{k_i}^{-1}r_{k_i} + u_{k_i}^T r_{k_i} + g_{k_i}^T v_{k_i} \\ &\leq -\kappa \|u_{k_i}\|^2 + \kappa_1 \|u_{k_i}\| \|c_{k_i}\| + \kappa_2 \|c_{k_i}\| \\ &\leq \|u_{k_i}\| (-\kappa \epsilon + \kappa_1 h_{k_i} + \frac{\kappa_2}{\epsilon} h_{k_i}), \end{aligned} \quad (3.5)$$

for some constants $\kappa_1, \kappa_2 > 0$. We define $\epsilon_1 := \min\{M_h, \frac{\epsilon^2 \kappa}{2(\kappa_1 \epsilon + \kappa_2)}\}$. Then if $h(x_{k_i}) \leq \epsilon_1$, we can obtain, by (3.5),

$$g_{k_i}^T d_{k_i} \leq -\frac{\epsilon \kappa}{2} \chi(x_{k_i}) \leq -\frac{\epsilon^2 \kappa}{2}.$$

Hence (3.2) holds with $\epsilon_2 := \frac{\epsilon^2 \kappa}{2}$.

For Algorithm 2 and 5, we have that $\hat{u}_k = -W_k^{-1}g_k + W_k^{-1}A_k(A_k^T A_k)^{-1}A_k^T g_k + W_k^{-1}r_k$. Then

$$u_k = P_k \hat{u}_k = -W_k^{-1}g_k + W_k^{-1}A_k(A_k^T W_k^{-1}A_k)^{-1}A_k^T W_k^{-1}g_k + P_k W_k^{-1}r_k.$$

with P_k given by (1.3b).

Hence

$$g_k^T u_k = -u_k^T W_k u_k + u_k^T W_k P_k W_k^{-1} r_k. \quad (3.6)$$

Similarly, we can get that for $k_i \notin \bar{\mathcal{R}}$

$$\begin{aligned} g_{k_i}^T d_{k_i} &= g_{k_i}^T u_{k_i} + g_{k_i}^T v_{k_i} \\ &= -u_{k_i}^T W_{k_i} u_{k_i} + u_{k_i}^T W_{k_i} P_{k_i} W_{k_i}^{-1} r_{k_i} + g_{k_i}^T v_{k_i} \\ &\leq -\kappa \|u_{k_i}\|^2 + \hat{\kappa}_1 \|u_{k_i}\| \|c_{k_i}\| + \hat{\kappa}_2 \|c_{k_i}\| \\ &\leq \|u_{k_i}\| (-\kappa \epsilon + \hat{\kappa}_1 h_{k_i} + \frac{\hat{\kappa}_2}{\epsilon} h_{k_i}), \end{aligned}$$

for some constants $\hat{\kappa}_1, \hat{\kappa}_2 > 0$. We can also define $\epsilon_1 := \min\{M_h, \frac{\epsilon^2 \kappa}{2(\hat{\kappa}_1 \epsilon + \hat{\kappa}_2)}\}$. Then if $h(x_{k_i}) \leq \epsilon_1$, we can obtain, by (3.6),

$$g_{k_i}^T d_{k_i} \leq -\frac{\epsilon \kappa}{2} \chi(x_{k_i}) \leq -\frac{\epsilon^2 \kappa}{2}.$$

Hence (3.2) holds with $\epsilon_2 := \frac{\epsilon^2 \kappa}{2}$.

For Algorithm 3 and 6, we have that

$$\begin{aligned} \hat{u}_k &= -W_k^{-1}g_k + W_k^{-1}A_k(A_k^T W_k^{-1}A_k)^{-1}A_k^T W_k^{-1} \nabla f_k \\ &\quad - W_k^{-1}A_k(A_k^T W_k^{-1}A_k)^{-1}c_k + W_k^{-1}r_k. \end{aligned}$$

Then

$$\begin{aligned} u_k &= P_k \hat{u}_k \\ &= (I - A_k[A_k^T A_k]^{-1}A_k^T) \hat{u}_k \\ &= -W_k^{-1}g_k + W_k^{-1}A_k(A_k^T W_k^{-1}A_k)^{-1}A_k^T W_k^{-1}g_k \\ &\quad - W_k^{-1}A_k(A_k^T W_k^{-1}A_k)^{-1}c_k + A_k(A_k^T A_k)^{-1}c_k + P_k W_k^{-1}r_k \\ &= \hat{u}_k - W_k^{-1}r_k - v_k + P_k W_k^{-1}r_k \end{aligned}$$

with P_k given by (1.3a). Hence

$$\begin{aligned} u_k^T W_k u_k &= u_k^T W_k \hat{u}_k - u_k^T r_k - u_k^T W_k v_k + u_k^T W_k P_k W_k^{-1} r_k \\ &= -u_k^T g_k - u_k^T W_k v_k + u_k^T W_k P_k W_k^{-1} r_k. \end{aligned}$$

So

$$g_k^T u_k = -u_k^T W_k u_k - u_k^T W_k v_k + u_k^T W_k P_k W_k^{-1} r_k. \quad (3.7)$$

Similarly, there exist constants ϵ_1 and ϵ_2 such that (3.2) holds. $\square$

We note that the result (3.2) is precisely that established in LEMMA 2 of [10]. Therefore, under Assumptions G we can apply the same analysis as in [10] to obtain the following global convergence results.

Theorem 3.1. Suppose Assumptions G hold. Then

$$\lim_{k \rightarrow \infty} h(x_k) = 0 \quad \text{and} \quad \liminf_{k \rightarrow \infty} \chi(x_k) = 0.$$

4 Local convergence

Now we turn to establish a superlinear convergence for our algorithm. As the filter approach will suffer from the Maratos effect, hence it is necessary to overcome Maratos effect by solving second order correction step $\hat{d}_k$ in our algorithm.

Assume that the sequence $\{x_k\}$ generated by the proposed algorithm converges to a local solution x^* , we give the following assumption, which is used in the local convergence analysis.

Assumptions L. The sequences $\{x_k\}, \{\lambda_k\}$ generated by Algorithm are contained in a convex set Ω and the following properties hold:

- L1. f and c are twice continuously differentiable in a neighborhood Ω of x^* .
- L2. There exists a constant $M_h > 0$ such that $k \notin \mathcal{R}$ whenever $h_k \leq M_h$.
- L3. $A(x^*)$ is full row rank.
- L4. W_k is uniformly positive definite on the null space of A_k^T , and W^* is positive definite on the null space of $A(x^*)^T$. The sequence $\{W_k\}$ is uniformly bounded.
- L5. W_k satisfies that

$$\lim_{k \rightarrow \infty} \frac{\|[W_k - \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_{k+1})]d_k\|}{\|d_k\|} = 0. \quad (4.1)$$

Although the proof of the following results are essentially similar to those presented in [11], we include it here for completeness. We define an exact penalty function

$$\phi_\omega(x) := f(x) + \omega h(x), \quad (4.2)$$

and use the penalty function in local convergence analysis. The local linearized approximation of the penalty function is

$$q_\omega(x_k, d) = f(x_k) + g_k^T d + \frac{1}{2} d^T W_k d + \omega \|A_k^T d_k + c(x_k)\|. \quad (4.3)$$

Lemma 4.1. Suppose Assumptions L hold. Choose $\eta_k = O(\|d_k\|)$, then the correction vector $\hat{d}_k$ given in (2.2) satisfies

$$\|\hat{d}_k\| = O(\|d_k\|^2) \text{ and } c(x_k + d_k + \hat{d}_k) = o(\|d_k\|^2). \quad (4.4)$$

Proof. From (2.7) and $\eta_k = O(\|d_k\|)$, we have that

$$c(x_k + d_k) = c_k + A_k^T d_k + O(\|d_k\|^2) = \pi r_k + O(\|d_k\|^2) = O(\|d_k\|^2),$$

where $\pi > 0$. Hence, by (2.8), we can obtain that $\|\hat{d}_k\| = O(\|d_k\|^2)$. Similarly, we can get that

$$\begin{aligned} c(x_k + d_k + \hat{d}_k) &= c(x_k + d_k) + A(x_k + d_k)^T \hat{d}_k + O(\|\hat{d}_k\|^2) \\ &= [A(x_k + d_k) - A(x_k)]^T \hat{d}_k + O(\|\hat{d}_k\|^2) \\ &= o(\|\hat{d}_k\|) + O(\|\hat{d}_k\|^2) = o(\|d_k\|^2). \end{aligned}$$

□

Lemma 4.2. If Assumptions L hold, then there exists a constant $\tilde{\gamma}_\omega > 0$ such that, when $\omega \geq \tilde{\gamma}_\omega$,

$$q_\omega(x_k, 0) - q_\omega(x_k, d_k) \geq 0. \quad (4.5)$$

Proof. For Algorithm 1 and 4, (3.4) implies that

$$\begin{aligned} g_k^T d_k &= g_k^T u_k + g_k^T v_k \\ &= -u_k^T W_k u_k + g_k^T W_k^{-1} A_k (A_k^T W_k^{-1} A_k)^{-1} A_k^T W_k^{-1} r_k + u_k^T r_k + g_k^T v_k. \end{aligned} \quad (4.6)$$

From (4.3), we have for $\pi > 0$ that $q_\omega(x_k, 0) - q_\omega(x_k, d_k) = -g_k^T d_k - \frac{1}{2} d_k^T W_k d_k + \omega(\|c_k\| - \pi\|r_k\|)$. Hence

$$\begin{aligned} & q_\omega(x_k, 0) - q_\omega(x_k, d_k) \\ &= \frac{1}{2} u_k^T W_k u_k - \frac{1}{2} v_k^T W_k v_k - u_k^T W_k v_k - g_k^T W_k^{-1} A_k (A_k^T W_k^{-1} A_k)^{-1} A_k^T W_k^{-1} r_k - u_k^T r_k \\ & \quad - g_k^T v_k + \omega(\|c_k\| - \|r_k\|). \end{aligned} \quad (4.7)$$

Since $r_k = O(\|c_k\|)$, $v_k = O(\|c_k\|)$ and $u_k, v_k, W_k, \lambda_k^+$ are all bounded, we can get, by Assumption L4,

$$\begin{aligned} & q_\omega(x_k, 0) - q_\omega(x_k, d_k) \\ & \geq -\frac{1}{2} v_k^T W_k v_k - u_k^T W_k v_k - g_k^T W_k^{-1} A_k (A_k^T W_k^{-1} A_k)^{-1} A_k^T W_k^{-1} r_k - u_k^T r_k \\ & \quad - g_k^T v_k + \omega(\|c_k\| - \|r_k\|) \geq \omega(1-t)\|c_k\| - \kappa_3\|c_k\|, \end{aligned}$$

where $\kappa_3 > 0$. Let $\tilde{\gamma}_\omega := \frac{\kappa_3}{1-t}$. Then (4.5) holds for $\omega \geq \tilde{\gamma}_\omega$.

For Algorithm 2 and 5, (3.6) implies that

$$g_k^T d_k = g_k^T u_k + g_k^T v_k = -u_k^T W_k u_k + u_k^T W_k P_k W_k^{-1} r_k + g_k^T v_k. \quad (4.8)$$

Hence

$$\begin{aligned} & q_\omega(x_k, 0) - q_\omega(x_k, d_k) \\ &= \frac{1}{2} u_k^T W_k u_k - \frac{1}{2} v_k^T W_k v_k - u_k^T W_k v_k - u_k^T W_k P_k W_k^{-1} r_k - g_k^T v_k + \omega(\|c_k\| - \|r_k\|) \\ & \geq \omega(1-t)\|c_k\| - \hat{\kappa}_3\|c_k\|, \end{aligned} \quad (4.9)$$

where $\hat{\kappa}_3 > 0$. Let $\tilde{\gamma}_\omega := \frac{\hat{\kappa}_3}{1-t}$. Then (4.5) holds for $\omega \geq \tilde{\gamma}_\omega$.

For Algorithm 3 and 6, (3.7) implies that

$$g_k^T d_k = g_k^T u_k + g_k^T v_k = -u_k^T W_k u_k - u_k^T W_k v_k + u_k^T W_k P_k W_k^{-1} r_k + g_k^T v_k. \quad (4.10)$$

Hence

$$\begin{aligned} & q_\omega(x_k, 0) - q_\omega(x_k, d_k) \\ &= \frac{1}{2} u_k^T W_k u_k - \frac{1}{2} v_k^T W_k v_k - u_k^T W_k P_k W_k^{-1} r_k - g_k^T v_k + \omega(\|c_k\| - \|r_k\|). \end{aligned} \quad (4.11)$$

Similarly, we can also prove that there exists a $\tilde{\gamma}_\omega$ such that (4.5) holds for $\omega \geq \tilde{\gamma}_\omega$. $\square$

Lemma 4.3. Suppose Assumptions L hold. There exists a neighborhood $\mathcal{B}_1$ of x^* such that if (2.4) holds for $\alpha_{k,l} = 1$, then the trial step is accepted by the Armijo condition (2.5) for $x_k \in \mathcal{B}_1$.

Proof. If (2.4) holds, we have

$$h(x_k) < \tau^{-\frac{1}{s_h}} (-g_k^T d_k)^{\frac{s_f}{s_h}}. \quad (4.12)$$

Since $s_f > 2s_h$ and g_k is uniformly bounded, (4.12) implies that $h(x_k) = o(\|d_k\|^2)$. For Algorithm 1 and 4, (1.4b) and (2.2b) imply that

$$g_k^T \hat{d}_k = -g_k^T W_k^{-1} A_k (A_k^T W_k^{-1} A_k)^{-1} c(x_k + d_k) = \lambda_k^T c(x_k + d_k). \quad (4.13)$$

For Algorithm 2 and 5, (1.4a) and (2.2a) imply that

$$g_k^T \hat{d}_k = -g_k^T A_k (A_k^T A_k)^{-1} c(x_k + d_k) = \lambda_k^T c(x_k + d_k). \quad (4.14)$$

For Algorithm 3 and 6, (1.4c) and (2.2c) imply that

$$\begin{aligned} g_k^T \hat{d}_k &= -g_k^T W_k^{-1} A_k (A_k^T W_k^{-1} A_k)^{-1} c(x_k + d_k) \\ &= \lambda_k^T c(x_k + d_k) + O(\|c_k\|) \\ &= \lambda_k^T c(x_k + d_k) + o(\|d_k\|^2). \end{aligned} \quad (4.15)$$

Using second order Taylor expansions, (4.13)-(4.15) imply

$$\begin{aligned} g_k^T \hat{d}_k &= \lambda_k^T [c(x_k) + A(x_k)^T d_k + \frac{1}{2} d_k^T \nabla^2 c(x_k) d_k] + o(\|d_k\|^2) \\ &= \frac{1}{2} d_k^T \left[\sum_{i=1}^m \lambda_{k,i} \nabla^2 c_i(x_k) \right] d_k + o(\|d_k\|^2). \end{aligned} \quad (4.16)$$

By (4.6), (4.8), (4.10), $r_k = O(\|c_k\|)$ and $v_k = O(\|c_k\|)$, we can easily get

$$g_k^T d_k = -u_k^T W_k u_k + O(\|c_k\|) = -u_k^T W_k u_k + o(\|d_k\|^2). \quad (4.17)$$

Hence, by Assumption L5, (4.16) and (4.17), we have that

$$\begin{aligned} &f(x_k) - f(x_k + d_k + \hat{d}_k) + \beta g_k^T d_k \\ &= -g_k^T (d_k + \hat{d}_k) - \frac{1}{2} (d_k + \hat{d}_k)^T \nabla^2 f(x_k) (d_k + \hat{d}_k) + \beta g_k^T d_k + o(\|d_k\|^2) \\ &\stackrel{(4.16)}{=} (\beta - 1) g_k^T d_k - \frac{1}{2} d_k^T \left[\sum_{i=1}^m \lambda_{k,i} \nabla^2 c_i(x_k) \right] d_k - \frac{1}{2} d_k^T \nabla^2 f(x_k) d_k + o(\|d_k\|^2) \\ &\stackrel{(4.17)}{=} \left(\frac{1}{2} - \beta \right) u_k^T W_k u_k - u_k^T W_k v_k - \frac{1}{2} v_k^T W_k v_k + o(\|d_k\|^2) \\ &= \left(\frac{1}{2} - \beta \right) u_k^T W_k u_k + O(\|c_k\|) + o(\|d_k\|^2) \\ &= \left(\frac{1}{2} - \beta \right) u_k^T W_k u_k + o(\|d_k\|^2). \end{aligned} \quad (4.18)$$

From Assumption L4, $\beta \in (0, \frac{1}{2})$ and (4.18), we can conclude that the Armijo condition (2.5) holds for k sufficiently large. $\square$

Lemma 4.4. Suppose Assumptions L hold. There exist a neighborhood $\mathcal{B}_2 \subseteq \mathcal{B}_1$ and a constant $\gamma_\omega > 0$ such that for $\omega \geq \gamma_\omega$ and $x_k \in \mathcal{B}_2$,

$$\phi_\omega(x_k) - \phi_\omega(x_k + d_k + \hat{d}_k) \geq \frac{1 + \gamma_h}{2} [q_\omega(x_k, 0) - q_\omega(x_k, d_k)] \geq 0. \quad (4.19)$$

Proof. From (4.2), (4.16) and Lemma 4.1, we have that

$$\begin{aligned}
& \phi_\omega(x_k) - \phi_\omega(x_k + d_k + \hat{d}_k) \\
&= f_k + \omega \|c_k\| - f(x_k + d_k + \hat{d}_k) - \omega \|c(x_k + d_k + \hat{d}_k)\| \\
&= \omega \|c_k\| - g_k^T(d_k + \hat{d}_k) - \frac{1}{2}(d_k + \hat{d}_k)^T \nabla^2 f(x_k)(d_k + \hat{d}_k) + o(\|d_k\|^2) \\
&= \omega \|c_k\| - g_k^T d_k - \frac{1}{2} d_k^T \left[\sum_{i=1}^m \lambda_{k,i} \nabla^2 c_i(x_k) \right] d_k - \frac{1}{2} d_k^T \nabla^2 f(x_k) d_k + o(\|d_k\|^2) \\
&= \omega \|c_k\| - g_k^T d_k - \frac{1}{2} d_k^T W_k d_k + o(\|d_k\|^2) \\
&\stackrel{(4.17)}{=} \omega \|c_k\| + u_k^T W_k u_k - \frac{1}{2} d_k^T W_k d_k + O(\|c_k\|) + o(\|d_k\|^2) \\
&= \omega \|c_k\| + \frac{1}{2} u_k^T W_k u_k + O(\|c_k\|) + o(\|d_k\|^2). \tag{4.20}
\end{aligned}$$

We have, from (3.3), (4.7), (4.9) and (4.11), that

$$q_\omega(x_k, 0) - q_\omega(x_k, d_k) = \omega(\|c_k\| - \|r_k\|) + \frac{1}{2} u_k^T W_k u_k + O(\|c_k\|).$$

Hence, from Assumption L4, (4.19), $\gamma_h \in (0, 1)$ and $\|r_k\| = O(\|c_k\|)$, we obtain that

$$\begin{aligned}
& \phi_\omega(x_k) - \phi_\omega(x_k + d_k + \hat{d}_k) - \frac{1 + \gamma_h}{2} [q_\omega(x_k, 0) - q_\omega(x_k, d_k)] \\
&= \omega \|c_k\| + \frac{1}{2} u_k^T W_k u_k - \frac{1 + \gamma_h}{2} [\omega(\|c_k\| - \|r_k\|) + \frac{1}{2} u_k^T W_k u_k] + O(\|c_k\|) + o(\|d_k\|^2) \\
&= \frac{1 - \gamma_h}{2} \omega \|c_k\| + \frac{1 - \gamma_h}{4} u_k^T W_k u_k + O(\|c_k\|) + o(\|d_k\|^2) \\
&\geq \frac{1 - \gamma_h}{2} \omega \|c_k\| - \kappa_4 \|c_k\|,
\end{aligned}$$

which means for $\omega \geq \frac{2\kappa_4}{1 - \gamma_h}$, $\phi_\omega(x_k) - \phi_\omega(x_k + d_k + \hat{d}_k) \geq \frac{1 + \gamma_h}{2} [q_\omega(x_k, 0) - q_\omega(x_k, d_k)]$. Define $\gamma_\omega := \max\{\tilde{\gamma}_\omega, \frac{2\kappa_4}{1 - \gamma_h}\}$. From Lemma 4.2, we conclude that (4.19) holds for $\omega \geq \gamma_\omega$. $\square$

From [11], we know that the next three choices of step are also second order correction steps to d_k , and Lemma 4.4 is true for each of them.

$$\begin{aligned}
\bar{d}_k &= \sigma_k \hat{d}_k + d_{k+1} + \sigma_{k+1} \hat{d}_{k+1}, \\
\bar{d}_k &= \sigma_k \hat{d}_k + d_{k+1} + \sigma_{k+1} \hat{d}_{k+1} + d_{k+2} + \sigma_{k+2} \hat{d}_{k+2}, \\
\text{or } \bar{d}_k &= \sigma_k \hat{d}_k + d_{k+1} + \sigma_{k+1} \hat{d}_{k+1} + d_{k+2} + \sigma_{k+2} \hat{d}_{k+2} + d_{k+3} + \sigma_{k+3} \hat{d}_{k+3},
\end{aligned}$$

where $\sigma_k, \sigma_{k+1}, \sigma_{k+2}, \sigma_{k+3} \in \{0, 1\}$.

Lemma 4.5. Suppose Assumptions L hold. There exist some constants $\omega_1, \omega_2, \omega_3 > 0$ with

$$2\gamma_h \omega_2 < (1 - t)(1 + \gamma_h)(\omega_2 - \omega_1) - 2\gamma_f, \tag{4.21a}$$

$$2\omega_3 \geq (1 - t)(1 + \gamma_h)\omega_1 + [(1 - \gamma_h) + t(1 + \gamma_h)]\omega_2, \tag{4.21b}$$

such that $\omega_1, \omega_2, \omega_3 > \gamma_\omega$, where γ_ω is defined in Lemma 4.4.

Proof. We can find a constant ω_1 such that $\omega_1 > \gamma_\omega$. Next, we define $\omega_2 := \frac{(1-t)(1+\gamma_h)\omega_1 + 3\gamma_f}{(1-t)(1+\gamma_h) - 2\gamma_h}$, and

$$\omega_3 := (1 - \gamma_h)\omega_2 - \gamma_f, \tag{4.22}$$

where $t \in (0, \frac{1 - \gamma_h}{1 + \gamma_h})$. It is easy to verify that $\omega_2, \omega_3 \geq \omega_1 > \gamma_\omega$ and (4.21) hold. $\square$

We define the level set $L = \{x \in \mathcal{B}_2 : \phi_{\omega_3}(x) \leq \phi_{\omega_3}(x^*) + \delta\}$. Choose $\delta > 0$ such that for $x \in L$, we have $(h(x), f(x)) \notin \mathcal{F}_{K_1}$. Set K_2 as the first iteration satisfying $K_2 \geq K_1$ and $x_{K_2} \in L$. For $k \in N$, we define the filter building blocks

$$\mathcal{G}_k = \{(h, f) : h \geq (1 - \gamma_h)h(x_k) \text{ and } f \geq f(x_k) - \gamma_f h(x_k)\}.$$

Define the index sets $I_{k_1}^{k_2} = \{l = k_1, \dots, k_2 - 1 : l \in \mathcal{A}\}$ for $k_1 \leq k_2$. Hence, we have that

$$\mathcal{F}_{k_2} = \mathcal{F}_{k_1} \cup \bigcup_{l \in I_{k_1}^{k_2}} \mathcal{G}_l. \quad (4.23)$$

Lemma 4.6. Suppose Assumptions L hold. When $l \geq K_1$ with $h(x_l) > 0$, we have that

- (i) If $\phi_{\omega_2}(x_l) - \phi_{\omega_2}(x) \geq \frac{1 + \gamma_h}{2}[q_{\omega_2}(x_l, 0) - q_{\omega_2}(x_l, d_l)]$, then $(h(x), f(x)) \notin \mathcal{G}_l$.
- (ii) If $x \in L$ and $\phi_{\omega_2}(x_{K_2}) - \phi_{\omega_2}(x) \geq \frac{1 + \gamma_h}{2}[q_{\omega_2}(x_{K_2}, 0) - q_{\omega_2}(x_{K_2}, d_{K_2})]$, then $(h(x), f(x)) \notin \mathcal{F}_{K_2}$.

Proof. Firstly, we show that (i) is true. When $\phi_{\omega_2}(x_l) - \phi_{\omega_2}(x) \geq \frac{1 + \gamma_h}{2}[q_{\omega_2}(x_l, 0) - q_{\omega_2}(x_l, d_l)]$, by (2.8) and (4.3), we can get

$$\begin{aligned} \phi_{\omega_2}(x_l) - \phi_{\omega_2}(x) &\geq \frac{1 + \gamma_h}{2}[q_{\omega_2}(x_l, 0) - q_{\omega_2}(x_l, d_l)] \\ &= \frac{1 + \gamma_h}{2}\{[q_{\omega_1}(x_l, 0) - q_{\omega_1}(x_l, d_l)] + (\omega_2 - \omega_1)[h(x_l) - \|A_l^T d_l + c_l\|]\} \\ &\geq \frac{1 + \gamma_h}{2}(\omega_2 - \omega_1)[h(x_l) - \|r_l\|] \\ &\geq \frac{(1 - t)(1 + \gamma_h)}{2}(\omega_2 - \omega_1)h(x_l). \end{aligned} \quad (4.24)$$

If $f(x) < f(x_l) - \gamma_f h(x_l)$, then $(h(x), f(x)) \notin \mathcal{G}_l$. Otherwise, we have $f(x) \geq f(x_l) - \gamma_f h(x_l)$. From (4.2), (4.21a), (4.24) and $h(x_l) > 0$, we can obtain that

$$\begin{aligned} h(x_l) - h(x) &\geq \frac{(1 - t)(1 + \gamma_h)}{2\omega_2}(\omega_2 - \omega_1)h(x_l) + \frac{1}{\omega_2}[f(x) - f(x_l)] \\ &\geq \frac{(1 - t)(1 + \gamma_h)}{2\omega_2}(\omega_2 - \omega_1)h(x_l) - \frac{\gamma_f}{\omega_2}h(x_l) > \gamma_h h(x_l), \end{aligned} \quad (4.25)$$

which implies $(h(x), f(x)) \notin \mathcal{G}_l$. Hence, (i) holds.

Next, we show that (ii) is true. Since $x \in L$, we have $(h(x), f(x)) \notin \mathcal{F}_{K_1}$. From (4.23), we only need to show that for all $l \in I_{K_1}^{K_2}$, $(h(x), f(x)) \notin \mathcal{G}_l$. By (2.8) and (4.3), we can get

$$\begin{aligned} \phi_{\omega_2}(x_{K_2}) - \phi_{\omega_2}(x) &\geq \frac{1 + \gamma_h}{2}[q_{\omega_1}(x_{K_2}, 0) - q_{\omega_1}(x_{K_2}, d_{K_2})] + (\omega_2 - \omega_1)h(x_{K_2}) \\ &\geq \frac{1 + \gamma_h}{2}(\omega_2 - \omega_1)[h(x_{K_2}) - \|r_{K_2}\|] \\ &\geq \frac{(1 - t)(1 + \gamma_h)}{2}(\omega_2 - \omega_1)h(x_{K_2}). \end{aligned} \quad (4.26)$$

From (4.2), (4.21b), (4.26) and the definition of K_2 and $l < K_2$, we can obtain that

$$\begin{aligned} \phi_{\omega_3}(x_l) &> \phi_{\omega_3}(x_{K_2}) \\ &= \phi_{\omega_2}(x_{K_2}) + (\omega_3 - \omega_2)h(x_{K_2}) \\ &\geq \phi_{\omega_2}(x) + [\omega_3 - \frac{(1 - t)(1 + \gamma_h)}{2}\omega_1 - \frac{(1 - \gamma_h) + t(1 + \gamma_h)}{2}\omega_2]h(x_{K_2}) \\ &\geq \phi_{\omega_2}(x). \end{aligned} \quad (4.27)$$

If $f(x) < f(x_l) - \gamma_f h(x_l)$, then $(h(x), f(x)) \notin \mathcal{G}_l$. Otherwise, we have $f(x) \geq f(x_l) - \gamma_f h(x_l)$. This, along with (4.2), (4.22) and (4.27), implies that

$$\begin{aligned}
h(x) &= \frac{1}{\omega_2}[\phi_{\omega_2}(x) - f(x)] \\
&< \frac{1}{\omega_2}[\phi_{\omega_3}(x_l) - f(x)] \\
&= \frac{1}{\omega_2}[f(x_l) + \omega_3 h(x_l) - f(x)] \\
&\leq \frac{\omega_3 + \gamma_f}{\omega_2} h(x_l) \\
&= (1 - \gamma_h) h(x_l),
\end{aligned} \tag{4.28}$$

which implies $(h(x), f(x)) \notin \mathcal{G}_l$. Hence, (ii) holds. $\square$

Now, we can get the main local convergence theorem which can be proved in a way similar to Theorem 4.7 in [11].

Theorem 4.1. Suppose Assumptions L hold. For k sufficiently large, the sequence $\{x_k\}$ converges to x^* with a superlinear rate.

5 Numerical results

Numerical experiments on the method given in this paper have been performed on a personal computer with Intel Pentium Dual CPU (2.00GHz). In this section, we present the numerical results of the proposed algorithm and give comparison with the method considered in [3]. In [3], Byrd, Curtis, and Nocedal proposed an Inexact Newton line search algorithm (Algorithm INS) without filter technique for large-scale equality constrained optimization. Obviously, the two methods are essentially comparable.

The numerical results have been obtained by running our algorithm on the set of 23 equality constrained problems. These problems, as others from Table 3 in [3] (those labeled with a (*)), were taken from the CUTEr [1, 8] collection. The starting point supplied with the problem was used. All codes were written in Matlab code. All attempts to solve the test problems were limited to a maximum of 1000 iterations. The computation terminates when the following stopping criteria is satisfied $\|\nabla_x \mathcal{L}(x_k, \lambda_k)\|_1 + \|c_k\|_1 \leq 10^{-5}$. We set the parameters in the algorithm as follows: $\beta = 0.3$, $\gamma_h = 0.01$, $\gamma_f = 0.5$, $\tau = 1$, $\nu = 10^4$, $s_h = 1.5$, $s_f = 3.2$, $\gamma_\alpha = 0.5$ and $\varsigma = 0.5$, which seemed to work reasonably well for a broad class of problems.

Detailed numerical results can be found in Tables 5.2-5.4, where the headers are described in Table 5.1. For comparison, we list the iterative number of Algorithm INS [3], which is devoted by Iter.. We can see from these tables that the run for each problem terminated successfully and Algorithm A is feasible and effective. In most cases, the number of iterations and function evaluations are few and it is competitive when compared with Algorithm INS [3]. It is particularly comforting to view the results as one can see that Algorithm A approximates a fast Newton method for the same equality constrained problems.

References

- [1] I.Bongarta, A.R.Conn, N.I.M.Gould, and Ph.L.Toint, CUTE:Constrained and unconstrained testing environment, ACM Transaction on Mathematical Software, 21(1995)123-160.

- [2] R.H.Byrd, F.E.Curtis and J.Nocedal, An inexact SQP method for equality constrained optimization, *SIAM J. Optim.*, 19(2008)351-369.
- [3] R.H.Byrd, F.E.Curtis and J.Nocedal, An inexact Newton method for nonconvex equality constrained optimization. *Math. Program., Ser. A*, 122(2010)273-299
- [4] C.M.Chin and R.Fletcher, On the global convergence of an SLP-filter algorithm takes EQP steps, *Math. Program.*, 96(2003)161-177.
- [5] R.Fletcher, N.I.M.Gould, S.Leyffer, P.L.Toint and A.Wächter, A global convergence of a trust region SQP-filter algorithm for general nonlinear programming, *SIAM J. Optim.*, 13(2002)635-660.
- [6] R.Fletcher and S.Leyffer, Nonlinear programming without a penalty function, *Math. Program.*, 91(2002)239-269.
- [7] R.Fontecilla, Local convergence of secant methods for nonlinear constrained optimization, *SIAM J. Numer. Anal.*, 25(1988)692-712.
- [8] N.I.M.Gould, D.Orban, and Ph.l.Toint, CUTEr and sifdec:A constrained and unconstrained testing environment, revisited, *ACM Trans.Math.Softw.*, 29(2003)-373-394.
- [9] E.Karas, A.Ribeiro, C.Sogastizábal and M.Solodov, A bundle-filter method for nonsmooth convex constrained optimization, *Math. Program.*, 116(2009)297-320.
- [10] A.Wächter and L.T.Biegler, Line search filter methods for nonlinear programming: Motivation and Global convergence, *SIAM J. Comput.*, 16(2005)1-31.
- [11] A.Wächter and L.T.Biegler, Line search filter methods for nonlinear programming: Local convergence, *SIAM J. Optim.*, 6(2005)32-48.
- [12] X.Zhu and D.Pu, A restoration-free filter SQP algorithm for equality constrained optimization, *Appl. Math. Comput.*, 219(2013)6016-6029.
- [13] Z.Wang and D.Zhu, Projected affine-scaling interior-point Newton's method with line search filter for box constrained optimization, *Appl. Math. Comput.*, 230(2014)484-495.