J. Oper. Res. Soc. China (2016) 4:385–396
DOI 10.1007/s40305-016-0122-6

Two-Phase-SQP Method with Higher-Order Convergence Property

Suvra Kanti Chakraborty[1] · Geetanjali Panda[1]

Received: 11 June 2015 / Revised: 20 January 2016 / Accepted: 24 February 2016 /
Published online: 2 April 2016

Abstract We propose a two-phase-SQP (Sequential Quadratic Programming) algorithm for equality-constrained optimization problem. In this paper, an iteration process is developed, and at each iteration, two quadratic sub-problems are solved. It is proved that, under some suitable assumptions and without computing further higher-order derivatives, this iteration process achieves higher-order local convergence property in comparison to Newton-SQP scheme. Theoretical advantage and a note on l_1 merit function associated to the method are provided.

Keywords Constrained optimization · Newton-SQP method · Cubic-order convergence

Mathematics Subject Classification 90C30 · 90C55

1 Introduction

Sequential quadratic programming (SQP) method is one of the most successful methods for solving constrained nonlinear optimization problems. This is an iterative procedure which generates a sequence of points (not necessarily feasible points), obtained by solving quadratic programming sub problems, and converges to the Karush-Kuhn-Tucker (KKT) point. This idea was first proposed by Wilson [1] in 1963. Since then SQP method has been studied extensively by many researchers (see

✉ Geetanjali Panda
geetanjali@maths.iitkgp.ernet.in

Suvra Kanti Chakraborty
suvrakanti@maths.iitkgp.ernet.in

[1] Department of Mathematics, Indian Institute of Technology Kharagpur, Kharagpur 761302, India

[2–8]). The readers may see Boggs [9], Gould et al. [10], Schittkowski et al. [11] for some good reviews on SQP algorithms, discussed so far. Some SQP methods use convex quasi-Newton approximation which makes the algorithm slow in case of large scale problems, whereas some other SQP methods employ the exact Hessian of the Lagrangian, which are Newton like methods. Under suitable assumptions the Newton version of the SQP algorithm converges to the minimum point without the use of line search and additional parameters. The Newton-SQP framework is the same as solving a Newton system derived from the KKT conditions, in which case it is often difficult to choose an initial iterate, close enough to the true solution to guarantee the convergence of the algorithm. These SQP methods, discussed so far, at most show quadratic or superlinear local convergence property. For the past few decades, researchers have shown their interest (see [12–16]) in iterative algorithms with higher-order convergence property. Again we notice that equality-constrained optimization emerges as a special branch of interest in constrained optimization theory (see [17,18]).

In this paper, we suggest a two-phase-SQP method for equality-constrained optimization problems, which provides local cubic-order convergence under some comfortable assumptions. We have also discussed the conditions for the associated line search method to the scheme, under the l_1 merit function.

Section 2 explains the Newton-SQP method. The two-phase-SQP is proposed in Sect. 3 which is followed by the convergence analysis in Sect. 4. We propose a note on l_1 merit function of the new scheme in Sect. 5. Numerical examples are provided in Sect. 6, and finally some concluding remarks are provided in Sect. 7.

2 Background: Existing Newton-SQP Method

Consider the following optimization problem with equality constraints.

$$\text{(EP):}\quad \text{Min } f(x) \tag{2.1a}$$

$$\text{s.t. } h(x) = 0, \tag{2.1b}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions. The motivation behind the local SQP approach is to model (EP) at the current iterate x_k by a quadratic programming subproblem, then use the minimizer of the subproblem to define a new iterate x_{k+1}.

Lagrangian function for the problem (EP) is $\mathscr{L}(x,\lambda) = f(x) - \lambda^{\mathrm{T}} h(x),\ \lambda \in \mathbb{R}^m$. Denote $A(x) = [\nabla h_1(x), \nabla h_2(x), \cdots, \nabla h_m(x)]^{\mathrm{T}}$ where $h_i(x)$ is the ith component of the vector $h(x)$. The KKT system for (EP) is

$$P(x,\lambda) \triangleq \begin{bmatrix} \nabla f(x) - A(x)^{\mathrm{T}}\lambda \\ h(x) \end{bmatrix} = 0. \tag{2.2}$$

Any solution (x^*, λ^*) of (EP) for which $A(x)$ has full rank, satisfies (2.2). System (2.2) consists of $n + m$ equations in $n + m$ unknowns x and λ. Let $P'(x_k, \lambda_k)$ be the first-order Frechet derivative of P at (x_k, λ_k). The next iterate (x_{k+1}, λ_{k+1}) is given by

$x_{k+1} = x_k + \widehat{x}_k$ and $\lambda_{k+1} = \lambda_k + \widehat{\lambda}_k$, where $(\widehat{x}_k, \widehat{\lambda}_k)$ is the value of $(\alpha, \beta) \in \mathbb{R}^n \times \mathbb{R}^m$, satisfying the following system:

$$P'(x_k, \lambda_k)\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = -P(x_k, \lambda_k),$$

which is same as

$$\begin{bmatrix}\nabla^2_{xx}\mathscr{L}_k & -\,A_k^{\mathrm{T}}\\ A_k & 0\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}-\nabla f_k + A_k^{\mathrm{T}}\lambda_k\\ -h_k\end{bmatrix}, \tag{2.3}$$

where $\mathscr{L}_k \triangleq \mathscr{L}(x_k, \lambda_k),\ f_k \triangleq f(x_k),\ A_k \triangleq A(x_k)$.

The Newton step generated from the iterate (x_k, λ_k) is thus given by

$$\begin{bmatrix}x_{k+1}\\ \lambda_{k+1}\end{bmatrix} = \begin{bmatrix}x_k\\ \lambda_k\end{bmatrix} + \begin{bmatrix}\widehat{x}_k\\ \widehat{\lambda}_k\end{bmatrix}, \tag{2.4}$$

where $\widehat{x}_k$ and $\widehat{\lambda}_k$ can be obtained by solving (2.3). If the following assumptions are true, then (2.3) has unique solution (see [19]).

Assumptions

(A-1) The constraint Jacobian matrix $A(x)$ has full rank.

(A-2) The matrix $\nabla^2_{xx}\mathscr{L}(x, \lambda)$ is positive definite on the tangent space of the constraints, i.e., $d^{\mathrm{T}}\nabla^2_{xx}\mathscr{L}(x, \lambda)d > 0$ for $d \in \{d \in \mathbb{R}^n \mid A(x)d = 0, d \neq 0\}$.

Consider the following quadratic programming problem as an approximate model of (EP) at (x_k, λ_k):

$$\begin{aligned}(\mathrm{QP}_k):\quad & \mathrm{Min}_{s\in\mathbb{R}^n}\ \ f_k + \nabla f_k^{\mathrm{T}} s + \frac{1}{2}s^{\mathrm{T}}\nabla^2_{xx}\mathscr{L}_k\, s\\ & \text{s.t.}\ \ A_k s + h_k = 0.\end{aligned}$$

The Lagrange function of (QP$_k$) is

$$\mathscr{L}_{\mathrm{QP}} \triangleq f_k + \nabla f_k^{\mathrm{T}} s + \frac{1}{2}s^{\mathrm{T}}\nabla^2_{xx}\mathscr{L}_k\, s - l^{\mathrm{T}}(A_k s + h_k),$$

where l is the Lagrange multiplier of (QP$_k$). If the assumptions (A-1) and (A-2) hold, then (QP$_k$) has a unique solution (s_k, l_k), that satisfies the first-order KKT conditions of (QP$_k$), which are:

$$\nabla^2_{xx}\mathscr{L}_k s_k + \nabla f_k - A_k^{\mathrm{T}} l_k = 0, \tag{2.5a}$$

$$A_k s_k + h_k = 0. \tag{2.5b}$$

These two equations can be written in the following matrix form:

$$\begin{bmatrix}\nabla^2_{xx}\mathscr{L}_k & -A_k^{\mathrm{T}}\\ A_k & 0\end{bmatrix}\begin{bmatrix}s_k\\ l_k\end{bmatrix} = \begin{bmatrix}-\nabla f_k\\ -h_k\end{bmatrix}. \tag{2.6}$$

Subtracting $A_k^{\mathrm{T}}\lambda_k$ from both side of the first equation of (2.3), we have

$$\begin{bmatrix} \nabla_{xx}^2\mathscr{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} \widehat{x}_k \\ \lambda_k + \widehat{\lambda}_k \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -h_k \end{bmatrix},$$

i.e.,

$$\begin{bmatrix} \nabla_{xx}^2\mathscr{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} \widehat{x}_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -h_k \end{bmatrix}, \tag{2.7}$$

where $\lambda_{k+1} = \lambda_k + \widehat{\lambda}_{k+1}$.

Now the local SQP iterates are (x_{k+1}, λ_{k+1}), where $x_{k+1} = x_k + s_k$ and $\lambda_{k+1} = l_k$. If the assumptions (A-1) and (A-2) hold then $P'(x_k, \lambda_k)$ becomes a non singular matrix. Hence from (2.6) and (2.7) we have that $s_k = \widehat{x}_k$ and $l_k = \lambda_{k+1}$. In this way, the new iterate (x_{k+1}, λ_{k+1}) can be generated by solving (QP$_k$). Further it is proved in [9] that chosen (x_0, λ_0) close to the solution (x^*, λ^*) of (EP), then sequence $\{(x_k, \lambda_k)\}$ converges to (x^*, λ^*) quadratically.

In the next section, this concept is used to develop a new algorithm to solve (EP) with higher-order convergence property.

3 Two-Phase-SQP Technique

Let (s_k, l_k) be the solution of (QP$_k$), which is obtained using the process described in Sect. 2. l_k is the Lagrange multiplier at kth stage. Denote

$$x_{\bar{k}} = x_k + s_k \quad \text{and} \quad \lambda_{\bar{k}} = l_k.$$

At this new point $(x_{\bar{k}}, \lambda_{\bar{k}})$ find $A_{\bar{k}} \triangleq [\nabla h_1(x_{\bar{k}}), \nabla h_2(x_{\bar{k}}), \cdots, \nabla h_m(x_{\bar{k}})]^{\mathrm{T}}$ and $\nabla_{xx}^2\mathscr{L}_{\bar{k}} \triangleq \nabla_{xx}^2\mathscr{L}(x_{\bar{k}}, \lambda_{\bar{k}})$. Consider the following optimization problem (QP$_k$1) at (x_k, λ_k):

$$(\mathrm{QP}_k1)\text{:}\quad \mathrm{Min}_{s\in\mathbb{R}^n}\ f_k + \left(\nabla f_k^{\mathrm{T}} + \tfrac{1}{2}\lambda_k^{\mathrm{T}} B_k\right)s + \tfrac{1}{4}s^{\mathrm{T}}\left(\nabla_{xx}^2\mathscr{L}_k + \nabla_{xx}^2\mathscr{L}_{\bar{k}}\right) s \tag{3.1a}$$

$$\text{s.t. } \tfrac{1}{2}C_k s + h_k = 0, \tag{3.1b}$$

where $B_k = A_{\bar{k}} - A_k$, $C_k = A_{\bar{k}} + A_k$.

(QP$_k$1) is a convex quadratic programming problem on the tangent subspace $\{d \mid d^{\mathrm{T}}\nabla_{xx}^2\mathscr{L}(x, \lambda)d > 0, A(x)d = 0\}$. Assume that C_k is of full rank. The unique solution $(s_{\bar{k}}, l_{\bar{k}})$ of (QP$_k$1) satisfies the first-order KKT optimality condition of (QP$_k$1). $l_{\bar{k}}$ be the Lagrange multiplier of (QP$_k$1). Then from KKT optimality conditions

$$\frac{1}{2}\left(\nabla_{xx}^2\mathscr{L}_k + \nabla_{xx}^2\mathscr{L}_{\bar{k}}\right) s_{\bar{k}} + \nabla f_k + \frac{1}{2}B_k^{\mathrm{T}}\lambda_k - \frac{1}{2}C_k^{\mathrm{T}} l_{\bar{k}} = 0, \tag{3.2a}$$

$$\frac{1}{2}C_k s_{\bar{k}} + h_k = 0. \tag{3.2b}$$

Adding $\frac{1}{2}C_k^{\mathrm{T}}\lambda_k$ to both sides of (3.2a), we get

$$\frac{1}{2}\left(\nabla_{xx}^2\mathscr{L}_k+\nabla_{xx}^2\mathscr{L}_{\bar{k}}\right)s_{\bar{k}}+\nabla f_k+\frac{1}{2}B_k^{\mathrm{T}}\lambda_k-\frac{1}{2}C_k^{\mathrm{T}}(l_{\bar{k}}-\lambda_k)=\frac{1}{2}C_k^{\mathrm{T}}\lambda_k, \tag{3.3a}$$

$$\frac{1}{2}C_k s_{\bar{k}}+h_k=0. \tag{3.3b}$$

(3.3a) and (3.3b) can be written as the following matrix equation:

$$\begin{bmatrix}\frac{1}{2}(\nabla_{xx}^2\mathscr{L}_k+\nabla_{xx}^2\mathscr{L}_{\bar{k}}) & -\frac{1}{2}C_k^{\mathrm{T}}\\ \frac{1}{2}C_k & 0\end{bmatrix}\begin{bmatrix}s_{\bar{k}}\\ l_{\bar{k}}-\lambda_k\end{bmatrix}=\begin{bmatrix}-\nabla f_k+A_k^{\mathrm{T}}\lambda_k\\ -h_k\end{bmatrix}.$$

This is equivalent to

$$\frac{1}{2}\left(\begin{bmatrix}\nabla_{xx}^2\mathscr{L}_k & -A_k^{\mathrm{T}}\\ A_k & 0\end{bmatrix}+\begin{bmatrix}\nabla_{xx}^2\mathscr{L}_{\bar{k}} & -A_{\bar{k}}^{\mathrm{T}}\\ A_{\bar{k}} & 0\end{bmatrix}\right)\begin{bmatrix}s_{\bar{k}}\\ l_{\bar{k}}-\lambda_k\end{bmatrix}=\begin{bmatrix}-\nabla f_k+A_k^{\mathrm{T}}\lambda_k\\ -h_k\end{bmatrix},$$

i.e.,

$$\frac{1}{2}\left(P'(x_k,\lambda_k)+P'(x_{\bar{k}},\lambda_{\bar{k}})\right)\begin{bmatrix}s_{\bar{k}}\\ l_{\bar{k}}-\lambda_k\end{bmatrix}=-P(x_k,\lambda_k). \tag{3.4}$$

Denote

$$\widehat{x_{\bar{k}}}=s_{\bar{k}}\quad\text{and}\ \widehat{\lambda_{\bar{k}}}=l_{\bar{k}}-\lambda_k.$$

The iteration process discussed above can be summarized in the following two steps:

First step:

$$\begin{bmatrix}x_{\bar{k}}\\ \lambda_{\bar{k}}\end{bmatrix}=\begin{bmatrix}x_k\\ \lambda_k\end{bmatrix}+\begin{bmatrix}\widehat{x_k}\\ \widehat{\lambda_k}\end{bmatrix}, \tag{3.5a}$$

where ($\widehat{x_k}$,$\widehat{\lambda_k}$) is the value of (α,β) satisfying

$$P'(x_k,\lambda_k)\begin{bmatrix}\alpha\\ \beta\end{bmatrix}=-P(x_k,\lambda_k). \tag{3.5b}$$

Second step:

$$\begin{bmatrix}x_{k+1}\\ \lambda_{k+1}\end{bmatrix}=\begin{bmatrix}x_k\\ \lambda_k\end{bmatrix}+\begin{bmatrix}\widehat{x_{\bar{k}}}\\ \widehat{\lambda_{\bar{k}}}\end{bmatrix}, \tag{3.6a}$$

where ($\widehat{x_{\bar{k}}}$,$\widehat{\lambda_{\bar{k}}}$) is the value of (α,β) satisfying

$$\frac{1}{2}\left(P'(x_k,\lambda_k)+P'(x_{\bar{k}},\lambda_{\bar{k}})\right)\begin{bmatrix}\alpha\\ \beta\end{bmatrix}=-P(x_k,\lambda_k). \tag{3.6b}$$

Since $\lambda_{k+1} = \lambda_k + \widehat{\lambda_{\bar{k}}}$, so from (3.4) and (3.6a), $\lambda_{k+1} = \lambda_k + (l_{\bar{k}} - \lambda_k) = l_{\bar{k}}$. Hence the new iterate (x_{k+1}, λ_{k+1}) can be generated by solving the KKT optimality conditions given in (3.2a) or by its equivalent form given in (3.5) and (3.6).

We state the algorithm for the proposed two-phase-SQP (TP-SQP) scheme in its simplest form in the following algorithm:

Algorithm 1: **Algorithm for Two-Phase-SQP method**

```
Select an initial pair $(x_0, \lambda_0)$
do { Evaluate $f_k$, $\nabla f_k$, $\nabla^2_{xx}\mathscr{L}_k$, $h_k$ and $A_k$;
    Solve $(\mathrm{QP}_k)$ to obtain $s_k$ and $l_k$;
    Set $x_{\bar{k}} \leftarrow x_k + s_k$ and $\lambda_{\bar{k}} \leftarrow l_k$
    Evaluate $A_{\bar{k}}$ and $\nabla^2_{xx}\mathscr{L}_{\bar{k}}$;
    Solve $(\mathrm{QP}_k 1)$ to obtain $s_{\bar{k}}$ and $l_{\bar{k}} - \lambda_k$;
    Set $x_{k+1} \leftarrow x_k + s_{\bar{k}}$ and $\lambda_{k+1} \leftarrow l_{\bar{k}}$; }
Until (A Convergence test is not satisfied);
```

For solving the equality-constrained optimization problem, a good initial estimate x_0 for x^* can be used to obtain a good initial estimate λ_0 for the optimal multiplier vector λ^*, which may be considered as follows in the light of [9]:

$$\lambda_0 = \left[A(x_0)A(x_0)^{\mathrm{T}}\right]^{-1} A(x_0)\nabla f(x_0). \tag{3.7}$$

λ_0 can be made arbitrarily close to λ^* by choosing x_0 close to x^*.

4 Convergence Analysis

Theorem 4.1 *If P is thrice Frechet differentiable function and (x_0, λ_0) is sufficiently close to the solution (x^*, λ^*) of (EP), then sequence (x_k, λ_k) generated by Algorithm 1 converges cubically to (x^*, λ^*).*

Proof Let u_k and v_k are $m+n$ dimensional vectors, where $u_k \triangleq (x_k, \lambda_k)$ and $v_k \triangleq (x_{\bar{k}}, \lambda_{\bar{k}})$, respectively. Then the steps (3.5) and (3.6) can be written as follows.

$$v_k = u_k - \left[P'(u_k)\right]^{-1} P(u_k), \tag{4.1a}$$

$$u_{k+1} = u_k - 2\left[P'(u_k) + P'(v_k)\right]^{-1} P(u_k). \tag{4.1b}$$

We denote $\alpha = (x^*, \lambda^*)$. Here $P : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$. Using Taylor expansion of $P(\alpha)$ about u_k, we have

$$\begin{aligned} P(\alpha) = P(u_k) + P'(u_k)(\alpha - u_k) + \frac{1}{2!}P''(u_k)(\alpha - u_k)^2 \\ + \frac{1}{3!}P'''(u_k)(\alpha - u_k)^3 + \cdots \end{aligned} \tag{4.2}$$

Subtracting α from both the sides of (4.1b), and denoting $e_k = u_k - \alpha$, we have

$$\begin{aligned} e_{k+1} &= e_k - 2\left[P'(u_k) + P'(v_k)\right]^{-1} P(u_k), \\ \left[P'(u_k) + P'(v_k)\right] e_{k+1} &= \left[P'(u_k) + P'(v_k)\right] e_k - 2P(u_k). \end{aligned} \tag{4.3}$$

Since (x^*, λ^*) is the solution of (EP), so $P(x^*, \lambda^*) = 0$. From (4.2),

$$\begin{aligned} P(u_k) &= -P'(u_k)(\alpha - u_k) - \frac{1}{2!}P''(u_k)(\alpha - u_k)^2 \\ &\quad - \frac{1}{3!}P'''(u_k)(\alpha - u_k)^3 + \cdots \\ &= P'(u_k)e_k - \frac{1}{2!}P''(u_k)e_k^2 + \frac{1}{3!}P'''(u_k)e_k^3 + \cdots \end{aligned}$$

$$\begin{aligned} &[P'(u_k)]^{-1}P(u_k) \\ &= e_k - [P'(u_k)]^{-1}P''(u_k)\frac{1}{2!}e_k^2 + [P'(u_k)]^{-1}P'''(u_k)\frac{1}{3!}e_k^3 + \cdots \end{aligned} \tag{4.4}$$

Expanding $P'(v_k)$ with respect to u_k we get,

$$\begin{aligned} P'(v_k) &= P'(u_k) + P''(u_k)(v_k - u_k) + \frac{1}{2!}P'''(u_k)(v_k - u_k)^2 + \cdots \\ &= P'(u_k) - P''(u_k)\left[P'(u_k)^{-1}P(u_k)\right] + \frac{1}{2!}P'''(u_k)\left[P'(u_k)^{-1}P(u_k)\right]^2 + \cdots \\ &\quad \text{(From (4.1a))} \\ &= P'(u_k) - P''(u_k)\big[e_k - [P'(u_k)]^{-1}P''(u_k)\frac{1}{2!}e_k^2 \\ &\quad + [P'(u_k)]^{-1}P'''(u_k)\frac{1}{3!}e_k^3 + \cdots\big] + \frac{1}{2!}P'''(u_k)e_k^2 + \cdots \quad \text{(from (4.4))} \end{aligned} \tag{4.5}$$

The right hand side of (4.3) gives

$$\begin{aligned} \left[P'(u_k) + P'(v_k)\right] e_k - 2P(u_k) &= \frac{1}{2!}P''(u_k)\left[P'(u_k)\right]^{-1} P''(u_k)e_k^3 \\ &\quad + \frac{1}{2!}P'''(u_k)e_k^3 - \frac{2}{3!}P'''(u_k)e_k^3 + \cdots \end{aligned} \tag{4.6}$$

Using (4.6), Expression (4.3) can be written as

$$\left[P'(u_k) + P'(v_k)\right] e_{k+1} = \mathscr{B}_k e_k^3 + O(||e_k||^4), \tag{4.7}$$

where

$$\mathscr{B}_k = \left[\frac{1}{2!}P''(u_k)\left[P'(u_k)\right]^{-1}P''(u_k) + \frac{1}{3!}P'''(u_k)\right].$$

This shows that the Algorithm 1 has cubic-order convergence.

One may observe the following theoretical advantages of the TP-SQP scheme.

- Without computing higher-order derivatives, a higher-order convergence is achieved in comparison to Newton-SQP framework.
- TP-SQP scheme involves the same order of arithmetic operations as compared to Newton-SQP method.

One may associate a line search technique to each iteration of Algorithm 1 with the help of a merit function, which can decide whether a trial step should be accepted or not. In the next section, we limit our discussion to l_1 merit function for the proposed scheme. It has been discussed how to choose the penalty parameter μ that makes the chosen direction a descent one in every iteration.

5 A Note on l_1-Merit Function for the Proposed Scheme

The l_1-merit function is of the form $\phi_1 : \mathbb{R}^{n+1} \to \mathbb{R}$, as

$$\phi_1(x; \mu) = f(x) + \mu\|h(x)\|_1,$$

where μ is the penalty parameter, $\|.\|_1$ is l_1 norm. In TP-SQP method, the step length is of one unit length. The steplength $\alpha_k \in (0, 1]$ is associated if the following decrease condition holds:

$$\phi_1(x_k + \alpha_k p_k; \mu_k) \leqslant \phi_1(x_k; \mu_k) + \eta\alpha_k \mathscr{D}(\phi_1(x_k; \mu); p_k),$$

where $\eta \in (0, 1)$ and $\mathscr{D}(\phi_1(x_k; \mu); p_k)$ denotes the directional derivative of ϕ_1 in the direction p_k. This requirement is analogous to Armijo condition of unconstrained optimization problem provided that p_k is a descent direction, that is, $\mathscr{D}(\phi_1(x_k; \mu); p_k) \leqslant 0$. Descent direction holds if μ is chosen sufficiently large as shown in the following theorem:

Theorem 5.1 *For the TP-SQP scheme, if* $\max\,\{\|A_k s_{\bar{k}}\|_1, \|\frac{1}{2}(A_{\bar{k}} - A_k)s_{\bar{k}}\|_1\} \leqslant \|h_k\|_1$, *then* $\mu > 4\|\lambda_k\|_\infty + 2\|\lambda_{k+1}\|_\infty$ *implies*

$$\mathscr{D}(\phi_1(x_k; \mu); s_{\bar{k}}) \leqslant 0.$$

Proof

$$\begin{aligned}&\phi_1(x_k + \alpha s_{\bar{k}}; \mu) - \phi(x_k; \mu)\\ &= f(x_k + \alpha s_{\bar{k}}) - f_k + \mu\|h(x_k + \alpha s_{\bar{k}})\|_1 - \mu\|h_k\|_1\end{aligned}$$

$$
\begin{aligned}
&= (f_k + \alpha \nabla f_k^{\mathrm{T}} s_{\bar{k}} + \frac{1}{2}\alpha^2 s_{\bar{k}}^{\mathrm{T}} \nabla^2 f(x_k + t_1 s_{\bar{k}}) s_{\bar{k}}) - f_k \\
&\quad + \mu \| h_k + \alpha A_k s_{\bar{k}} + \frac{1}{2}\alpha^2 s_{\bar{k}}^{\mathrm{T}} \nabla^2 h(x_k + t_2 s_{\bar{k}}) s_{\bar{k}} \|_1 - \mu \| h_k \|_1 \\
&\qquad (\text{ where } t_1, t_2 \in (0, \alpha)) \\
&\leqslant \alpha \nabla f_k^{\mathrm{T}} s_{\bar{k}} + \mu \| h_k + \alpha A_k s_{\bar{k}} \|_1 - \mu \| h_k \|_1 + \gamma \alpha^2 \| s_{\bar{k}} \|_1^2 \\
&\qquad (\gamma \text{ is the upper bound of the second derivatives of } f \text{ and } h) \\
&= \alpha \nabla f_k^{\mathrm{T}} s_{\bar{k}} + \mu \| h_k + \alpha(-2h_k - A_{\bar{k}} s_{\bar{k}}) \|_1 - \mu \| h_k \|_1 + \gamma \alpha^2 \| s_{\bar{k}} \|_1^2 \\
&\qquad \left(\text{Since from (3.3b), } \ \frac{1}{2}(A_k + A_{\bar{k}}) s_{\bar{k}} + h_k = 0 \right) \\
&= \alpha \nabla f_k^{\mathrm{T}} s_{\bar{k}} + \mu \| (1-\alpha) h_k - \alpha(h_k + A_{\bar{k}} s_{\bar{k}}) \|_1 - \mu \| h_k \|_1 \\
&\quad + \gamma \alpha^2 \| s_{\bar{k}} \|_1^2 \\
&\leqslant \alpha \nabla f_k^{\mathrm{T}} s_{\bar{k}} + \mu (1-\alpha) \| h_k \|_1 + \mu \alpha \| h_k + A_{\bar{k}} s_{\bar{k}} \|_1 - \mu \| h_k \|_1 \\
&\quad + \gamma \alpha^2 \| s_{\bar{k}} \|_1^2 \quad (\text{for } 0 < \alpha \leqslant 1).
\end{aligned}
$$

This implies

$$
\lim_{\alpha \to 0} \frac{\phi_1(x_k + \alpha s_{\bar{k}}; \mu) - \phi_1(x_k; \mu)}{\alpha} \leqslant \nabla f_k^{\mathrm{T}} s_{\bar{k}} - \mu \| h_k \|_1 + \mu \| h_k + A_{\bar{k}} s_{\bar{k}} \|_1,
$$

i.e.,

$$
\mathscr{D}(\phi_1(x_k; \mu); s_{\bar{k}}) \leqslant \nabla f_k^{\mathrm{T}} s_{\bar{k}} - \mu \| h_k \|_1 + \mu \| h_k + A_{\bar{k}} s_{\bar{k}} \|_1.
$$

Multiplying $s_{\bar{k}}^{\mathrm{T}}$ in left to both sides of (3.2a) and using $l_{\bar{k}} = \lambda_{k+1}$, the above inequality becomes

$$
\begin{aligned}
&\mathscr{D}(\phi_1(x_k; \mu); s_{\bar{k}}) \\
&\leqslant -s_{\bar{k}}^{\mathrm{T}} \big(\frac{1}{2} (\nabla_{xx}^2 \mathscr{L}_k + \nabla_{xx}^2 \mathscr{L}_{\bar{k}}) \big) s_{\bar{k}} + s_{\bar{k}}^{\mathrm{T}} \big(\frac{1}{2} (A_k^{\mathrm{T}} + A_{\bar{k}}^{\mathrm{T}})(\lambda_{k+1} - \lambda_k) \big) \\
&+ s_{\bar{k}}^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_k - \mu \| h_k \|_1 + \mu \| h_k + A_{\bar{k}} s_{\bar{k}} \|_1.
\end{aligned}
$$

Using (3.3a), we get

$$
\begin{aligned}
\mathscr{D}(\phi_1(x_k; \mu); s_{\bar{k}}) &\leqslant -s_{\bar{k}}^{\mathrm{T}} \left(\frac{1}{2} \left(\nabla_{xx}^2 \mathscr{L}_k + \nabla_{xx}^2 \mathscr{L}_{\bar{k}} \right) \right) s_{\bar{k}} - h_k^{\mathrm{T}} \lambda_{k+1} + h_k^{\mathrm{T}} \lambda_k + (A_k s_{\bar{k}})^{\mathrm{T}} \lambda_k \\
&\quad - \mu \| h_k \|_1 + \mu \| \frac{1}{2} (A_{\bar{k}} - A_k) s_{\bar{k}} \|_1 \\
&\leqslant -s_{\bar{k}}^{\mathrm{T}} \left(\frac{1}{2} \left(\nabla_{xx}^2 \mathscr{L}_k + \nabla_{xx}^2 \mathscr{L}_{\bar{k}} \right) \right) s_{\bar{k}} + \| h_k \|_1 \| \lambda_{k+1} \|_\infty + \| h_k \|_1 \| \lambda_k \|_\infty \\
&\quad + \| A_k s_{\bar{k}} \|_1 \| \lambda_k \|_\infty - \mu \| h_k \|_1 + \frac{\mu}{2} \| h_k \|_1.
\end{aligned}
$$

Using the given condition max $\{\|A_k s_{\bar{k}}\|_1, \|\frac{1}{2}(A_{\bar{k}} - A_k)s_{\bar{k}}\|_1\} \leqslant \|h_k\|_1$, we get

$$\begin{aligned}\mathscr{D}(\phi_1(x_k;\mu); s_{\bar{k}}) &\leqslant -s_{\bar{k}}^{\mathrm{T}}\left(\frac{1}{2}(\nabla_{xx}^2\mathscr{L}_k + \nabla_{xx}^2\mathscr{L}_{\bar{k}})\right)s_{\bar{k}} \\ &\quad + \|h_k\|_1\left(\|\lambda_{k+1}\|_\infty + 2\|\lambda_k\|_\infty - \frac{\mu}{2}\right) \\ &\leqslant 0 \qquad (\text{for } \mu > 2\|\lambda_{k+1}\|_\infty + 4\|\lambda_k\|_\infty)\,.\end{aligned}$$

It follows that under the assumption stated, $s_{\bar{k}}$ will be a descent direction for ϕ if $\nabla_{xx}^2\mathscr{L}_{\bar{k}}$ is positive definite and $\mu > 2\|\lambda_{k+1}\|_\infty + 4\|\lambda_k\|_\infty$.

6 Numerical Examples

Following two examples are examined with the proposed algorithm, and the results are analyzed. The algorithm will terminate whenever the norm of the gradient of the Lagrangian comes close to zero.

Example 6.1

$$\begin{aligned}(\text{P1}): \min\ & x_1 + x_2 \\ \text{s.t. } & x_1^2 + x_2^2 - 2 = 0\end{aligned}$$

Example 6.2

$$\begin{aligned}(\text{P2}): \min\ & (\sqrt{3})^3 x_1 x_2 x_3 \\ \text{s.t. } & x_1^2 + x_2^2 + x_3^2 - 1 = 0\end{aligned}$$

(P1) and (P2) are solved in MATLAB R-2013(b) by Newton-SQP method and TP-SQP method with the tolerance limit is 10^{-15} and different initial points. Value of λ_0 is chosen using (3.7). The number of iterations by both the methods are provided in Table 1. Since the proposed scheme has local cubic-order convergence property, it is expected that the number of iterations of the proposed scheme to attain the solution must be less than the number of iterations of Newton-SQP scheme, which has quadratic convergence property. Table 1 certainly meets this expectation.

7 Conclusion

In this paper, we have proposed a local scheme to solve a nonlinear equality-constrained optimization problem with higher-order convergence property. Without computing higher-order derivatives, TP-SQP scheme achieves higher-order convergence property more quickly than Newton-SQP method. However, since we are solving two quadratic subproblems in each iteration, for the aspect of software implementation, TP-SQP scheme is more suitable than the Newton-SQP method for those optimization

Table 1 Comparison between the number of iterations for Newton-SQP and TP-SQP

	Minimizer $(x^*; \lambda^*)$	$(x_0; \lambda_0)$	Number of iterations (Newton-SQP)	Number of iterations (TP-SQP)
P1	$(-1, -1; 0.5)$	$(-0.8, -0.8; -0.625\,0)$	6	4
		$(-0.2, -0.3, -1.923\,1)$	10	5
		$(-1, 0; -0.5)$	8	5
		$(-0.9, -0.3; -0.667)$	8	4
P2	$\left(3^{-0.5}, 3^{-0.5}, 3^{-0.5}; -1.5\right)$	$(0.2, 0.3, 0.1; 0.334\,0)$	9	6
		$(0.4, 0.5, 0.6; 1.217\,4)$	6	3
		$(0.7, 0.6, 0.5; 1.488)$	6	3
		$(0.23, 0.34, 0.45; 0.739\,3)$	7	5

problems, where the former one takes almost half the number of iterations than the latter one. In future, in the light of TP-SQP scheme, one may propose schemes even having order of convergence more than three.

References

[1] Wilson, R.B.: A simplicial algorithm for concave programming. Ph.D. Dissertation. Harvard University (1963)

[2] Byrd, R.H., Curtis, F.E., Nocedal, J.: Infeasibility detection and SQP methods for nonlinear optimization. SIAM J. Optim. **20**(5), 2281–2299 (2010)

[3] Byrd, R.H., Nocedal, J.: An analysis of reduced Hessian methods for constrained optimization. Math. Program. **49**(1–3), 285–323 (1990)

[4] Conn, A.R., Gould, N.I., Toint, P.L.: Trust Region Methods, vol. 1. SIAM, Philadelphia (2000)

[5] Han, S.P.: Superlinearly convergent variable metric algorithms for general nonlinear programming problems. Math. Program. **11**(1), 263–282 (1976)

[6] Huang, M., Pu, D.: A trust-region SQP method without a penalty or a filter for nonlinear programming. J. Comput. Appl. Math. **281**, 107–119 (2015)

[7] Izmailov, A.F., Solodov, M.V.: A truncated SQP method based on inexact interior-point solutions of subproblems. SIAM J. Optim. **20**(5), 2584–2613 (2010)

[8] Mo, J., Zhang, K., Wei, Z.: A variant of SQP method for inequality constrained optimization and its global convergence. J. Comput. Appl. Math. **197**(1), 270–281 (2006)

[9] Boggs, P.T., Tolle, J.W.: Sequential quadratic programming. Acta Numer. **4**, 1–51 (1995)

[10] Gould, N., Orban, D., Toint, P.: Numerical methods for large-scale nonlinear optimization. Acta Numer. **14**, 299–361 (2005)

[11] Schittkowski, K., Yuan, Y.X.: Sequential Quadratic Programming Methods. Wiley Encyclopedia of Operations Research and Management Science, New York 147–224 (2011)

[12] Babajee, D., Dauhoo, M.: An analysis of the properties of the variants of Newtons method with third order convergence. Appl. Math. Comput. **183**(1), 659–684 (2006)

[13] Frontini, M., Sormani, E.: Third-order methods from quadrature formulae for solving systems of nonlinear equations. Appl. Math. Comput. **149**(3), 771–782 (2004)

[14] Golbabai, A., Javidi, M.: A third-order Newton type method for nonlinear equations based on modified homotopy perturbation method. Appl. Math. Comput. **191**(1), 199–205 (2007)

[15] Kou, J., Li, Y., Wang, X.: A modification of Newton method with third-order convergence. Appl. Math. Comput. **181**(2), 1106–1111 (2006)

[16] Sharma, J.R., Guha, R.K., Sharma, R.: An efficient fourth order weighted-newton method for systems of nonlinear equations. Numer. Algorithms **62**(2), 307–323 (2013)

[17] Grapiglia, G.N., Yuan, J., Yuan, Y.X.: A subspace version of the Powell–Yuan trust-region algorithm for equality constrained optimization. J. Oper. Res. Soc. China **1**(4), 425–451 (2013)

[18] Liu, X., Yuan, Y.: A sequential quadratic programming method without a penalty function or a filter for nonlinear equality constrained optimization. SIAM J. Optim. **21**(2), 545–571 (2011)

[19] Nocedal, J., Wright, S.J.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering, 2nd edn. Springer, New York (2006)