SUT Journal of Mathematics
Vol. 48, No. 2 (2012), 171–197

Inexact sequential quadratically constrained quadratic programming methods with nonmonotone line searches for convex programming

Atsushi Kato

(Received May 19, 2012; Revised October 23, 2012)

Abstract. In this paper, we present inexact sequential quadratically constrained quadratic programming (SQCQP) methods with nonmonotone line searches for the convex programming problem. Kato, Narushima and Yabe proposed the SQCQP method whose subproblem is solved inexactly. Their inexact SQCQP method uses a monotone line search strategy. To reduce the number of merit function evaluations and to accept the unit step size easier, we apply the nonmonotone line search to their inexact SQCQP method. We present the algorithms of the inexact SQCQP method with the nonmonotone line searches and prove their global and superlinear convergence properties. Moreover, we give some numerical experiments to investigate the numerical performance of our proposed methods.

AMS 2010 *Mathematics Subject Classification.* 65K05 90C30 90C55.

Key words and phrases. Convex programming, inequality constrained optimization, sequential quadratically constrained quadratic programming method, nonmonotone line search, global and superlinear convergence

§1. Introduction

We consider the following inequality constrained convex programming problem:

$$(1.1) \qquad \begin{cases} \min & f(x) \\ \text{s.t.} & g_i(x) \le 0 \quad (i = 1, \cdots, m), \end{cases}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ and $g_i : \mathbf{R}^n \to \mathbf{R}$ $(i = 1, \cdots, m)$ are twice continuously differentiable convex functions. For solving problem (1.1), we generally use the following iterative scheme:

$$x_{k+1} = x_k + \alpha_k d_k,$$

where the vector $x_k \in \mathbf{R}^n$ is the k-th approximate solution, the positive scalar α_k is a step size and the vector $d_k \in \mathbf{R}^n$ is a search direction at the k-th iteration.

The step size α_k is usually chosen such that the value of a merit function is monotonically decreasing within the framework of the line search strategy. To reduce the number of merit function evaluations and to accept the unit step size easier, the nonmonotone line search has been proposed by some authors [2, 3, 4, 5, 15, 19]. Dai [3] and Grippo, Lampariello and Lucidi [5] presented the Newton method with the nonmonotone line search for unconstrained optimization. Bonnans, Panier, Tits and Zhou [2] and Panier and Tits [15] considered the nonmonotone line search for constrained optimization problems. Yamashita and Yabe [19] studied sequential quadratic programming method with the nonmonotone line search and showed its global and superlinear convergence properties. Dai and Schittkowski [4] proposed the sequential quadratic programming method with the nonmonotone line search whose merit function is a differentiable augmented Lagrangian function.

The sequential quadratically constrained quadratic programming (SQCQP) method for problem (1.1) generates a search direction d_k by solving the following quadratically constrained quadratic programming subproblem at each iteration:

$$\text{(1.2)} \qquad \begin{cases} \min & \nabla f(x_k)^T d + \dfrac{1}{2} d^T B_k d \\ \text{s.t.} & g_i(x_k) + \nabla g_i(x_k)^T d + \dfrac{1}{2} d^T \nabla^2 g_i(x_k) d \le 0 \\ & (i = 1, \cdots, m), \end{cases}$$

where $\nabla f(x_k)$ and $\nabla g_i(x_k)$ are the k-th gradient vectors of f and g_i $(i = 1, \cdots, m)$, respectively. Moreover, the matrix B_k is symmetric positive definite and $\nabla^2 g_i(x_k)$ are the k-th Hessian matrices of g_i $(i = 1, \cdots, m)$. Subproblem (1.2) can be transformed to the second order cone programming problem [1] because subproblem (1.2) is a convex quadratically constrained quadratic programming. The SQCQP method has an advantage that it avoids the Maratos effect, that interrupts obtaining a superlinear convergent, while this method has a defect that subproblem (1.2) is not necessarily feasible.

The early studies of the SQCQP methods were done by Fukushima [6], Kruk and Wolkowicz [13, 14]. Some authors [7, 9, 10, 11, 17] proposed the feasible SQCQP methods, independently. Fukushima, Luo and Tseng [7] proposed the feasible SQCQP method whose subproblem is given by the form:

$$(1.3)\qquad \begin{cases} \min & \nabla f(x_k)^T d + \dfrac{1}{2} d^T B_k d \\ \text{s.t.} & g_i(x_k) + \nabla g_i(x_k)^T d + \dfrac{(\gamma_k)_i}{2} d^T \nabla^2 g_i(x_k) d \le 0 \\ & (i = 1, \cdots, m), \end{cases}$$

where $(\gamma_k)_i$ $(i = 1, \cdots, m)$ is either 0 or 1. Jian [9] proposed the norm-relaxed SQCQP method to obtain the feasibility. Jian, Hu, Tang and Zheng [10] presented the algorithm which used a feasible direction of descent. In this algorithm, the search direction is obtained by solving only one subproblem which contains a convex quadratic objective function and simple quadratic inequality constraints. Jian, Tang and Zheng [11] considered the SQCQP norm-relaxed algorithm of strongly sub-feasible directions. Solodov [17] presented the feasible quadratically constrained quadratic programming subproblem by introducing the slack variables. Kato, Narushima and Yabe [12] considered the SQCQP method whose feasible subproblem by Fukushima et. al [7] was solved inexactly. Tang and Jian [18] presented an SQCQP algorithm with an augmented Lagrangian line search function and made a brief comments on their SQCQP method with the nonmonotone line search.

In this paper, we apply the nonmonotone line search by Dai et al. [3, 4] to the inexact SQCQP method given in [12]. Since the SQCQP method uses the line search strategy, it is significant to consider the SQCQP method with the monotone line search and we can expect that this SQCQP method can reduce the number of merit function evaluations. Although we deal with the SQCQP methods for convex programming problem in theory, we can practically calculate the problem whose feasible region is a convex set (we do not require the convexity of the objective function) by our SQCQP methods.

This paper is organized as follows. In Section 2, we introduce the inexact SQCQP method, and review the monotone line search. We propose two inexact SQCQP methods with the nonmonotone line searches for convex problem (1.1) and prove the global and superlinear convergence of our methods in Section 3. In Section 4, we investigate numerical performance of our methods by preliminary numerical experiment.

In what follows, $\| \bullet \|$ and $\| \bullet \|_1$ denote the Euclid norm and the 1-norm for a vector, respectively. The matrix I denotes the identity matrix.

§2. Inexact SQCQP method with the monotone line search

In this section, we review the exact SQCQP method which is based on [7] and the inexact SQCQP method [12]. Throughout this paper, we assume that the set of optimal solutions is not empty and there exists an $\bar{x} \in \mathbf{R}^n$ satisfying the Slater condition $g_i(\bar{x}) < 0$ $(i = 1, \cdots, m)$. We call this point $\bar{x}$ the Slater point.

We first review how to obtain the feasible quadratically constrained quadratic programming subproblem. The feasibility of subproblem (1.3) depends on the parameter $\gamma_k = ((\gamma_k)_1, \cdots, (\gamma_k)_m)^T$. Therefore, this parameter must be chosen adequately so that the feasibility holds. For this purpose, Fukushima et al. [7] proposed the following algorithm. This algorithm is based on Lemma 2.1 of [7].

Algorithm Gamma

Step 1 Compute the parameters $(s_1)_k,\ (s_2)_k$ and $(s_3)_k$ as follows.

$$(s_1)_k = \max_{i \in I_{1,k}} \frac{g_i(x_k)}{g_i(x_k) - \psi g_i(\bar{x})},$$

$$(s_2)_k = \min\left\{ \min_{i \in I_{2,k}} \frac{g_i(x_k) - \psi g_i(\bar{x})}{\kappa_k},\ 1 \right\}, \tag{2.1}$$

$$(s_3)_k = \min\left\{ (s_2)_k,\ \min_{i \notin I_{2,k}} \frac{-2(\psi - \phi) g_i(\bar{x})}{\kappa_k} \right\}, \tag{2.2}$$

where $\phi \in [0, 1),\ \psi \in (\phi, 1),\ \kappa_k = (\bar{x} - x_k)^T \nabla^2 g_i(x_k)(\bar{x} - x_k)$ and the sets $I_{1,k}$ and $I_{2,k}$ are defined by

$$I_{1,k} = \{\ i \mid g_i(x_k) > 0\ \}, \qquad I_{2,k} = \{\ i \mid \phi g_i(\bar{x}) \le g_i(x_k)\ \}.$$

Here, if there is no $i \in I_{1,k}$, then we set $(s_1)_k := -\infty$ and if $\kappa_k = 0$, then we set $+\infty$ for the fractions in (2.1) and (2.2).

Step 2 Determine the parameter γ_k by using the following rule:

$$\begin{cases} (s_2)_k \le 2(s_1)_k & \to\ (\gamma_k)_i = 0,\ i = 1, \cdots, m, \\ (s_3)_k \le 2(s_1)_k < (s_2)_k & \to\ (\gamma_k)_i = \begin{cases} 1, & i \in I_{2,k}, \\ 0, & i \notin I_{2,k}, \end{cases} \\ (s_3)_k > 2(s_1)_k & \to\ (\gamma_k)_i = 1,\ i = 1, \cdots, m. \end{cases}$$

It is important to choose a suitable Slater point $\bar{x}$. By the relationships among $(s_1)_k$, $(s_2)_k$ and $(s_3)_k$, we can observe that subproblem (1.3) easily becomes quadratically constrained quadratic programming subproblem (1.2) when $g_i(\bar{x})$ $(i = 1, \cdots, m)$ is sufficiently small.

Kato, Narushima and Yabe [12] proposed the inexact SQCQP method. They relaxed the optimality conditions of subproblem (1.3) as follows:

$$
\begin{aligned}
(2.3)\qquad & \|\nabla f(x_k) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x_k) + H_k(\lambda) d\| && \leq && \varepsilon_1 \|d\|,\\
(2.4)\qquad & |\lambda_i \{g_i(x_k) + \nabla g_i(x_k)^T d + \frac{(\gamma_k)_i}{2} d^T \nabla^2 g_i(x_k) d\}| && \leq && \varepsilon_2 \|d\|^2,\\
(2.5)\qquad & g_i(x_k) + \nabla g_i(x_k)^T d + \frac{(\gamma_k)_i}{2} d^T \nabla^2 g_i(x_k) d && \leq && 0,\\
(2.6)\qquad & \lambda_i && \geq && 0,\\
& (i = 1, \cdots, m),
\end{aligned}
$$

where $\lambda_k = ((\lambda_k)_1, \cdots, (\lambda_k)_m)^T$ is the k-th Lagrange multiplier for the inequality constraint and the symmetric positive definite matrix $H_k(\lambda)$ is given by

$$H_k(\lambda) = B_k + \sum_{i=1}^{m} \lambda_i \{(\gamma_k)_i \nabla^2 g_i(x_k)\}.$$

Moreover, the parameters ε_1 and ε_2 are positive constants and we introduce how to choose these parameters later on. We find the pair (d, λ) which satisfies conditions (2.3)–(2.6) at each iteration. When $\varepsilon_1 = \varepsilon_2 = 0$, conditions (2.3)–(2.6) become the exact optimality conditions of subproblem (1.3). In this case, we call the method the exact SQCQP method, which is identical to the algorithm by [7].

We note that when $d = 0$ holds, conditions (2.3)–(2.6) reduce to the KKT conditions of problem (1.1):

$$
\begin{aligned}
(2.7)\qquad & \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) && = && 0, &&\\
(2.8)\qquad & \lambda_i g_i(x) && = && 0, && (i = 1, \cdots, m),\\
(2.9)\qquad & g_i(x) && \leq && 0, && (i = 1, \cdots, m),\\
(2.10)\qquad & \lambda_i && \geq && 0, && (i = 1, \cdots, m).
\end{aligned}
$$

We call a point (x, λ) which satisfies the above conditions the KKT point. Since problem (1.1) is a convex programming problem, the KKT point becomes an optimal solution. Thus, by taking advantage of this relationship, we use $\|d_k\| \leq \varepsilon$ for a small positive number ε as a stopping criterion to solve problem (1.1).

To establish global convergence, in the line search procedure, we use the following penalty merit function:

$$\theta_r(x) = f(x) + r p(x),$$

where a penalty parameter $r > 0$ is updated at each iteration and $p(x)$ is a penalty function defined by

$$p(x) = \max\{0, g_1(x), \cdots, g_m(x)\}.$$

In the monotone line search in [12], we find a step size α which satisfies

$$\theta_{r_k}(x_k + \alpha d_k) \leq \theta_{r_k}(x_k) - \nu\alpha d_k^T G_k(\lambda_{k+1}) d_k, \tag{2.11}$$

where $\nu \in (0, 1/2)$ is a constant and the symmetric positive definite matrix $G_k(\lambda_{k+1})$ is given by

$$G_k(\lambda_{k+1}) = B_k + \frac{1}{2}\sum_{i=1}^{m}(\lambda_{k+1})_i\{(\gamma_k)_i \nabla^2 g_i(x_k)\}.$$

This step size α guarantees to reduce the value of the merit function monotonically.

Now we introduce the algorithm of the inexact SQCQP method with monotone line search. By setting $\varepsilon_1 = \varepsilon_2 = 0$, this algorithm becomes the algorithm of the exact SQCQP method with monotone line search.

Algorithm ISQCQP

Step 0 Give an initial point x_0, a Slater point $\bar{x}$ and an initial penalty parameter $r_{-1} > 0$. Choose parameters $\varepsilon_1,\ \varepsilon_2,\ \omega > 0,\ \phi \in [0, 1),\ \psi \in (\phi, 1),\ \nu \in (0, 1/2),\ \tau \in (0, 1)$ and a tolerance $\varepsilon > 0$. Set $k = 0$.

Step 1 Having $x_k,\ \bar{x},\ \phi$ and ψ, calculate parameters γ_k by using Algorithm Gamma.

Step 2 Choose $n \times n$ symmetric positive definite matrix B_k. Having x_k, B_k and γ_k, find the pair (d_k, λ_{k+1}) which satisfies conditions (2.3)–(2.6).

Step 3 If $\|d_k\| \leq \varepsilon$ holds, then stop; otherwise, go to Step 4.

Step 4 If $r_k < \|\lambda_{k+1}\|_1 + \omega$, then set $r_k = \|\lambda_{k+1}\|_1 + \omega$; otherwise, set $r_k = r_{k-1}$.

Step 5 Find a step size α_k^{LS} by using Algorithm LS:

Algorithm LS

Step 5.1 Set $\ell = 0$.

Step 5.2 If the integer ℓ satisfies

$$\theta_{r_k}(x_k + \tau^\ell d_k) \leq \theta_{r_k}(x_k) - \nu\tau^\ell d_k^T G_k(\lambda_{k+1}) d_k, \tag{2.12}$$

then set $\alpha_k^{LS} = \tau^\ell$ and stop; otherwise, go to Step 5.3.

Step 5.3 Set $\ell := \ell + 1$, and return to Step 5.2.

Step 6 Set $x_{k+1} = x_k + \alpha_k^{LS} d_k$ and $k := k+1$. Return to Step 1.

We need the condition $r_k \geq \|\lambda_{k+1}\|_1$ to obtain a descent search direction. This is the reason why we set the rule of Step 4 in Algorithm ISQCQP.

To obtain the global convergence of Algorithm ISQCQP, we make Assumption Global given below.

Assumption Global

(G1) There exist positive constants ϑ_1 and ϑ_2 such that

$$\vartheta_1 \|v\|^2 \;\leq\; v^T B_k v \;\leq\; \vartheta_2 \|v\|^2$$

holds for all $v \in \mathbf{R}^n$ and any $k \geq 0$.

(G2) The parameters ε_1 and ε_2 are chosen such that

$$\left(\frac{1}{2} - \nu\right)\vartheta_1 \quad > \quad \varepsilon_1 + m\varepsilon_2.$$

(G3) The sequence $\{x_k\}$ generated by ISQCQP is bounded.

(G4) The sequence $\{d_k\}$ generated by ISQCQP is bounded.

(G5) The Hessian matrices $\nabla^2 f(x)$ and $\nabla^2 g_i(x)$ $(i = 1, \cdots, m)$ are Lipschitz continuous on a convex and bounded set $\mathcal{L}$ containing the sequences $\{x_k\}$ and $\{x_k + d_k\}$, i.e., there exists a positive constant ζ such that

$$\begin{aligned} \|\nabla^2 f(u) - \nabla^2 f(v)\| &\leq \zeta \|u - v\|, \\ \|\nabla^2 g_i(u) - \nabla^2 g_i(v)\| &\leq \zeta \|u - v\|, \quad (i = 1, \cdots, m), \end{aligned}$$

for all u and $v \in \mathcal{L}$.

Remark 1. When $\varepsilon_1 = \varepsilon_2 = 0$, i.e., we consider the exact SQCQP method, we do not need assumption (G4) (see Lemma 3.3 of [7]).

Under Assumption Global, we obtain the following global convergence property of Algorithm ISQCQP. This theorem was proved in Section 3 of [12].

Theorem 1. *Suppose that Assumption Global holds. Then Algorithm ISQCQP either terminates at an optimal solution of problem* (1.1) *or generates an infinite sequence* $\{x_k\}$ *of which every accumulation point is an optimal solution of problem* (1.1).

Furthermore, we make the following additional assumptions to obtain the superlinear convergence property of Algorithm ISQCQP.

Assumption Local

(L1) The problem (1.1) has a unique optimal solution x_*.

(L2) The linear independence constraint qualification holds at x_*, i.e., $\nabla g_i(x_*)$ $(i \in I(x_*))$ are linearly independent, where $I(x_*) = \{\ i \mid g_i(x_*) = 0\}$.

(L3) The second order sufficient condition holds at x_*, i.e.,

$$v^T \nabla_x^2 L(x_*, \lambda_*) v > 0$$

for any nonzero vector $v \in \mathbf{R}^n$ which satisfies

$$\nabla g_i(x_*)^T v = 0, \quad (i \in I(x_*)),$$

where λ_* is a Lagrangian multiplier associated with x_*.

(L4) The strict complementarity condition holds at x_*, i.e.,

$$\begin{aligned} g_i(x_*) = 0,\ (\lambda_*)_i > 0, &\qquad \text{for } i \in I(x_*), \\ g_i(x_*) < 0,\ (\lambda_*)_i = 0, &\qquad \text{for } i \notin I(x_*). \end{aligned}$$

(L5) The following holds:

$$\nabla_x L(x_k, \lambda_{k+1}) + H_k(\lambda_{k+1}) d_k = o(\|d_k\|). \tag{2.13}$$

(L6) The sequence $\{B_k\}$ satisfies

$$\{\nabla^2 f(x_k) - B_k\} d_k = o(\|d_k\|).$$

The next theorem implies that subproblem (1.3) becomes a quadratically constrained quadratic programming problem (1.2) when x_k is very close to the optimal solution, that we can avoid the Maratos effect and that the sequence $\{x_k\}$ generated by Algorithm ISQCQP converges superlinearly to x_*. This theorem was proved in Lemma 4.1, Lemma 4.2 and Theorem 4.3 in [12].

Theorem 2. *Let $\{x_k\}$ be the infinite sequence generated by Algorithm ISQCQP. Then the following holds.*

(i) *Parameters $(\gamma_k)_i$ given by Algorithm Gamma are set to* 1 *for $i = 1, \cdots, m$ and sufficiently large k.*

(ii) *If Assumption Global and assumption* (L6) *hold, then the unit step size $\alpha_k = 1$ is accepted for sufficiently large k.*

(iii) *If Assumptions Global and Local hold, then the sequence $\{x_k\}$ converges superlinearly to x_*.*

§3. Nonmonotone ISQCQP method and its convergence

In this section, we propose inexact SQCQP methods with nonmonotone line searches. Our nonmonotone line searches are based on Dai et al. [3, 4], reduces the number of merit function evaluations and accepts the unit step size easier.

Instead of condition (2.11), we propose the following nonmonotone condition

$$\theta_{r_k}(x_k + \alpha d_k) \leq \max_{0\leq j\leq M(k)}\{\theta_{r_k}(x_{k-j})\} - \nu\alpha d_k^T G_k(\lambda_{k+1})d_k, \tag{3.1}$$

where $\bar{M}$ is a positive integer and an integer $M(k)$ is defined by $M(0) = 0$ and $M(k) = \min\{M(k-1)+1, \bar{M}\}$ $(k \geq 1)$, and

$$\max_{0\leq j\leq M(k)}\{\theta_{r_k}(x_{k-j})\} = \max_{0\leq j\leq M(k)}\{f(x_{k-j}) + r_k p(x_{k-j})\}.$$

Note that the same penalty parameter r_k is used in $\theta_{r_k}(x_{k-j})$ for $j = 0, \cdots, M(k)$ and that condition (3.1) is equivalent to condition (2.11) when $\bar{M} = 0$. Furthermore, we propose the following alternative nonmonotone condition

$$\theta_{r_k}(x_k + \alpha d_k) \leq \varphi_{r_k}(x_k) - \nu\alpha d_k^T G_k(\lambda_{k+1})d_k, \tag{3.2}$$

where

$$\varphi_{r_k}(x_k) = \max_{0\leq j\leq M(k)}\{f(x_{k-j})\} + r_k p(x_k).$$

It is notable that condition (3.2) is weaker than condition (2.11) and stronger than condition (3.1).

Now we describe the algorithms of two inexact SQCQP methods with the nonmonotone line searches.

Algorithm NISQCQP1

Step 0 Give an initial point x_0, a Slater point $\bar{x}$ and an initial penalty parameter $r_{-1} > 0$. Choose parameters $\varepsilon_1,\ \varepsilon_2,\ \omega > 0,\ \phi \in [0,1),\ \psi \in (\phi, 1),\ \nu \in (0, 1/2),\ \tau \in (0,1)$, a positive integer $\bar{M} > 0$ and a tolerance $\varepsilon > 0$. Set $k = 0$.

Step 1 Having $x_k,\ \bar{x},\ \phi$ and ψ, calculate parameters γ_k by using Algorithm Gamma.

Step 2 Choose $n \times n$ symmetric positive definite matrix B_k. Having x_k, B_k and γ_k, find the pair (d_k, λ_{k+1}) which satisfies conditions (2.3)–(2.6).

Step 3 If $\|d_k\| \leq \varepsilon$ holds, i.e., the pair (x_k, λ_{k+1}) approximately satisfies the KKT conditions (2.7)–(2.10), then stop; otherwise, go to Step 4.

Step 4 If $r_k < \|\lambda_{k+1}\|_1 + \omega$, then set $r_k = \|\lambda_{k+1}\|_1 + \omega$; Otherwise, set $r_k = r_{k-1}$.

Step 5 Find a step size α_k^{N1} by using Algorithm NLS1:

Algorithm NLS1

Step 5.1 Set $\ell = 0$.

Step 5.2 If the integer ℓ satisfies

$$\theta_{r_k}(x_k + \tau^\ell d_k) \le \max_{0 \le j \le M(k)} \{\theta_{r_k}(x_{k-j})\} - \nu \tau^\ell d_k^T G_k(\lambda_{k+1}) d_k, \tag{3.3}$$

then set $\alpha_k^{N1} = \tau^\ell$ and stop; otherwise, go to Step 5.3.

Step 5.3 Set $\ell := \ell + 1$, and return to Step 5.2.

Step 6 Set $x_{k+1} = x_k + \alpha_k^{N1} d_k$ and $k := k + 1$. Return to Step 1.

Algorithm NISQCQP2

Step 0 Give an initial point x_0, a Slater point $\bar{x}$ and an initial penalty parameter $r_{-1} > 0$. Choose parameters ε_1, ε_2, $\omega > 0$, $\phi \in [0, 1)$, $\psi \in (\phi, 1)$, $\nu \in (0, 1/2)$, $\tau \in (0, 1)$, a positive integer $\bar{M} > 0$ and a tolerance $\varepsilon > 0$. Set $k = 0$.

Step 1 Having x_k, $\bar{x}$, ϕ and ψ, calculate parameters γ_k by using Algorithm Gamma.

Step 2 Choose $n \times n$ symmetric positive definite matrix B_k. Having x_k, B_k and γ_k, find the pair (d_k, λ_{k+1}) which satisfies conditions (2.3)–(2.6).

Step 3 If $\|d_k\| \le \varepsilon$ holds, i.e., the pair (x_k, λ_{k+1}) approximately satisfies the KKT conditions (2.7)–(2.10), then stop; otherwise, go to Step 4.

Step 4 If $r_k < \|\lambda_{k+1}\|_1 + \omega$, then set $r_k = \|\lambda_{k+1}\|_1 + \omega$; Otherwise, set $r_k = r_{k-1}$.

Step 5 Find a step size α_k^{N2} by using Algorithm NLS2:

Algorithm NLS2

Step 5.1 Set $\ell = 0$.

Step 5.2 If the integer ℓ satisfies

$$\theta_{r_k}(x_k + \tau^\ell d_k) \le \varphi_{r_k}(x_k) - \nu \tau^\ell d_k^T G_k(\lambda_{k+1}) d_k, \tag{3.4}$$

then set $\alpha_k^{N2} = \tau^\ell$ and stop; otherwise, go to Step 5.3.

Step 5.3 Set $\ell := \ell + 1$, and return to Step 5.2.

Step 6 Set $x_{k+1} = x_k + \alpha_k^{N2} d_k$ and $k := k + 1$. Return to Step 1.

It is notable that these algorithms correspond to the algorithm of the exact SQCQP method with the nonmonotone line search when $\varepsilon_1 = \varepsilon_2 = 0$ and that Algorithms NISQCQP1 and NISQCQP2 are same except for Step 5.

Now we prove the global convergence properties of Algorithms NISQCQP1 and NISQCQP2. From now on, we assume $d_k \neq 0$ for any k and we give some lemmas to show that the sequence $\{x_k\}$ generated by Algorithms NISQCQP1 or NISQCQP2 converges to an optimal solution of problem (1.1), respectively. The first lemma means that the search direction d_k generated by Step 2 of Algorithm NISQCQP1 or NISQCQP2 becomes a descent search direction at the k-th iteration for the merit function.

Lemma 1. *The pair* (d_k, λ_{k+1}) *satisfies*

$$D(\theta_{r_k}(x_k); d_k) \leq -d_k^T G_k(\lambda_{k+1}) d_k - (r_k - \|\lambda_{k+1}\|_1) p(x_k) + (\varepsilon_1 + m\varepsilon_2)\|d_k\|^2,$$

where the scalar $D(\theta_r(x); d)$ *denotes the directional derivative of function* $\theta_r(x)$ *at* x *in a direction* d*. Moreover, under assumptions* (G1) *and* (G2)*, the following holds*

$$D(\theta_{r_k}(x_k); d_k) \leq -\frac{1}{2}\vartheta_1 \|d_k\|^2.$$

This lemma can be proved in the same way as the proofs of Lemmas 3.1 and 3.2 in [12]. The second lemma guarantees that there exist a step size α_k^{N1} which satisfies condition (3.1) and a step size α_k^{N2} which satisfies condition (3.2).

Lemma 2. *Suppose that assumptions* (G1) *and* (G2) *hold. Then, for any* k*, the following hold:*

(i) *Algorithm NLS1 terminates at a finite number of iterations.*

(ii) *Algorithm NLS2 terminates at a finite number of iterations.*

Proof. (i) From Lemma 3.3 of [12], there exists the minimum nonnegative integer ℓ_k ($\alpha_k^{LS} = \tau^{\ell_k}$) which satisfies condition (2.12) at each k. Since this integer ℓ_k also satisfies condition (3.3), there exists the minimum nonnegative integer $\bar{\ell}_k$ ($\alpha_k^{N1} = \tau^{\bar{\ell}_k}$, $\bar{\ell}_k \leq \ell_k$) which satisfies condition (3.3) at each k. Therefore, (i) is proved.

(ii) We can prove (ii) in the same way as (i). □

The third lemma insists that the sequence $\{r_k\}$ of the penalty parameter is bounded. This lemma can be proved in the same way as Lemma 3.5 of [12].

Lemma 3. *Suppose that assumptions* (G1)–(G4) *hold. Then each sequence* $\{\lambda_{k+1}\}$ *generated by Algorithms NISQCQP1 and NISQCQP2 is bounded and there exist a positive constant* r_* *and a positive integer* $\bar{k}$ *such that*

$$r_k = r_* \quad \text{for all } k \ \geq \ \bar{k}. \tag{3.5}$$

The next lemma guarantees that the step sizes α_k^{N1} and α_k^{N2} are uniformly bounded away from zero.

Lemma 4. *Suppose that Assumption Global holds. Then,* $\alpha_k^{N1} \geq \alpha_*$ *and* $\alpha_k^{N2} \geq \alpha_*$ *hold for all* k *and a positive constant* α_* *defined by*

$$\alpha_* \equiv \tau \min\left\{\frac{\vartheta_1}{(\bar{r}+1)\mu}, 1\right\},$$

where μ *is a positive constant such that*

$$\|\nabla^2 f(x)\| \leq \mu \quad \text{and} \quad \|\nabla^2 g_i(x)\| \leq \mu \ \ (i = 1, \cdots, m) \ \text{ for all } x \in \mathcal{L}.$$

Proof. From the proof of Lemma 2, for all k, we obtain

$$\alpha_k^{N1} \geq \alpha_k^{N2} \geq \alpha_k^{LS}.$$

Furthermore, for all k, we have that

$$\alpha^{LS} \geq \alpha_*$$

from Lemma 3.6 in [12]. Thus, we get that

$$\alpha_k^{N1} \geq \alpha_k^{N2} \geq \alpha_k^{LS} \geq \alpha_*.$$

Therefore, the proof is complete. □

By using Lemmas 1–4, we obtain the following result.

Lemma 5. *Suppose that Assumption Global is satisfied. Then each sequence* $\{\|d_k\|\}$ *generated by Algorithms NISQCQP1 and NISQCQP2 satisfies*

$$\liminf_{k\to\infty} \|d_k\| = 0.$$

Proof. (i) We consider the case of Algorithm NISQCQP1. Since

$$\max_{0\leq j\leq M(k)}\{\theta_{r_k}(x_{k-j})\} = \max_{k-M(k)\leq j\leq k}\{\theta_{r_k}(x_j)\} \tag{3.6}$$

holds for any $k \geq \bar{M}$, we have, for any $k \geq \hat{k} = \max\{\bar{k},\ \bar{M}\}$, that

$$\begin{aligned} \theta_{r_*}(x_{k+1}) &\leq \max_{0\leq j\leq M(k)}\{\theta_{r_*}(x_{k-j})\} - \nu\alpha_k^{N1} d_k^T G_k(\lambda_{k+1}) d_k \\ &\leq \max_{0\leq j\leq M(k)}\{\theta_{r_*}(x_{k-j})\} - \nu\alpha_*\vartheta_1\|d_k\|^2 \end{aligned} \tag{3.7}$$

from condition (3.1), assumption (G1) and Lemmas 3 and 4.

We prove (i) by contradiction. For this purpose, we assume that there exists a positive constant η_1 such that

$$\|d_k\| \geq \eta_1 > 0$$

for any $k \geq \hat{k}$. It follows from (3.6) and (3.7) that

$$\begin{aligned} \theta_{r_*}(x_{k+1}) &\leq \max_{0\leq j\leq M(k)}\{\theta_{r_*}(x_{k-j})\} - \nu\alpha_*\vartheta_1\|d_k\|^2 \\ &\leq \max_{0\leq j\leq M(k)}\{\theta_{r_*}(x_{k-j})\} - \nu\alpha_*\vartheta_1\eta_1^2 \\ &= \max_{k-M(k)\leq j\leq k}\{\theta_{r_*}(x_j)\} - \nu\alpha_*\vartheta_1\eta_1^2 \\ &\leq \max_{k-\bar{M}\leq j\leq k}\{\theta_{r_*}(x_j)\} - \nu\alpha_*\vartheta_1\eta_1^2 \qquad (3.8) \end{aligned}$$

for any $k \geq \hat{k}$. Thus, we conclude from (3.8) that $\theta_{r_*}(x_k)$ tends to $-\infty$. However, this contradicts that $\{\theta_{r_*}(x_k)\}$ is bounded below from assumption (G3). Therefore, (i) is proven.

(ii) We show the case of Algorithm NISQCQP2. In the same way as (i), we have

$$\begin{aligned} \theta_{r_*}(x_{k+1}) &= f(x_{k+1}) + r_*p(x_{k+1}) \\ &\leq \varphi_{r_*}(x_k) - \nu\alpha_k d_k^T G_k(\lambda_{k+1})d_k \\ &\leq \max_{0\leq j\leq M(k)}\{f(x_{k-j})\} + r_*p(x_k) - \nu\alpha_*\vartheta_1\|d_k\|^2. \qquad (3.9) \end{aligned}$$

We prove (ii) by contradiction as well as (i). For this purpose, we assume that there exists a positive constant η_2 such that

$$\|d_k\| \geq \eta_2 > 0$$

for any $k \geq \hat{k}$. It follows from (3.9) that

$$\begin{aligned} \theta_{r_*}(x_{k+1}) &\leq \max_{0\leq j\leq M(k)}\{f(x_{k-j})\} + r_*p(x_k) - \nu\alpha_*\vartheta_1\eta_2^2 \\ &= \max_{k-M(k)\leq j\leq k}\{f(x_j)\} + r_*p(x_k) - \nu\alpha_*\vartheta_1\eta_2^2 \\ &\leq \max_{k-\bar{M}\leq j\leq k}\{\theta_{r_*}(x_j)\} - \nu\alpha_*\vartheta_1\eta_2^2 \qquad (3.10) \end{aligned}$$

for any $k \geq \hat{k}$. From relationship (3.10), $\theta_{r_*}(x_k)$ tends to $-\infty$ and this contradicts that $\{\theta_{r_*}(x_k)\}$ is bounded below as the same way as (i). Thus, (ii) is shown. □

We obtain the following global convergence property from the above lemmas and the KKT conditions (2.7)–(2.10) of problem (1.1).

Theorem 3. *Suppose that Assumption Global holds. Let* $\{x_k\}$ *be the infinite sequence generated by Algorithm NISQCQP1 or NISQCQP2. Then there exists an accumulation point of the infinite sequence* $\{(x_k, \lambda_{k+1})\}$ *that is an optimal solution* x_* *and the corresponding multiplier* λ_* *of problem* (1.1).

Under Assumptions Global and Local, we obtain the following theorem which can be showed in the same way as Theorem 2.

Theorem 4. *Let* $\{x_k\}$ *be the infinite sequence generated by Algorithm NISQCQP1 or NISQCQP2. Then the following hold.*

(i) *Parameters* $(\gamma_k)_i$ *given by Algorithm Gamma are set to* 1 *for* $i = 1, \cdots, m$ *and sufficiently large* k.

(ii) *If Assumption Global and assumption* (L6) *hold, then the unit step size* $\alpha_k = 1$ *is accepted for sufficiently large* k.

(iii) *If Assumptions Global and Local hold, then the sequence* $\{x_k\}$ *converges superlinearly to* x_*.

§4. Numerical experiments

In this section, we investigate numerical performance of our algorithms by preliminary numerical experiments. Especially, to investigate numerical performance of the nonmonotone line searches, we change the parameters ν and $\bar{M}$ into various values. We solve the test problems HS100 in [8] and s394 in [16] because Jian et al. [11] used these problems in their numerical experiment and these problems have only inequality constraints and a convex feasible region. In our numerical experiment, we investigate the number of iterations, the number of merit function evaluations in Step 5 of Algorithms ISQCQP, NISQCQP1 and NISQCQP2, and the number of accepting a unit step size, respectively. However, we do not compare the methods in CPU time because it depends on the performance of software to find the pair (d_k, λ_{k+1}) in Step 2 of each algorithm. The program is written in Matlab.

We find a pair (d, λ) in Step 2 of each algorithm by the primal-dual interior point method [20] because we take advantage of conditions (2.3)–(2.6). The parameters are chosen as follows: $r_{-1} = 10$, $\omega = 5.0$, $\varepsilon = 10^{-5}$, $\phi = 0.50$, $\psi = 0.75$ and $\tau = 0.75$. Moreover, by considering assumptions (G1) and (L6), we use the nonsingular matrix B_k defined by

$$B_k = \nabla^2 f(x_k) + \delta_k I,$$

where δ_k is defined by $\delta_0 = 1$ and $\delta_k = t\|d_{k-1}\|$ $(k \geq 1)$, and t is a positive constant (we choose t later on). In addition, by considering assumption (G2) i.e., by considering the relationships $0.50(1/2-\nu)\delta_k > \varepsilon_{1,k}$ and $0.50(1/2-\nu)\delta_k/m > \varepsilon_{2,k}$, we use parameters $\varepsilon_{1,k} = 0.49(1/2-\nu)\delta_k$ and $\varepsilon_{2,k} = 0.49(1/2-\nu)\delta_k/m$ instead of ε_1 and ε_2. We update parameters δ_k, $\varepsilon_{1,k}$ and $\varepsilon_{2,k}$ in Step 2 of each algorithm. To solve subproblem (1.3), we use the following practical relaxed conditions:

$$\begin{aligned}
&(4.1) & \|\nabla f(x_k) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x_k) + H_k(\lambda)d\| &\leq \max\{\varepsilon_{1,k}\|d\|,\ \bar{\varepsilon}\},\\
&(4.2) & |\lambda_i\{g_i(x_k) + \nabla g_i(x_k)^T d + \frac{(\gamma_k)_i}{2} d^T \nabla^2 g_i(x_k) d\}| &\leq \max\{\varepsilon_{2,k}\|d\|^2,\ \bar{\varepsilon}\},\\
&(4.3) & g_i(x_k) + \nabla g_i(x_k)^T d + \frac{(\gamma_k)_i}{2} d^T \nabla^2 g_i(x_k) d &\leq \bar{\varepsilon},\\
&(4.4) & \lambda_i &\geq 0,\\
& & (i = 1, &\cdots, m),
\end{aligned}$$

where $\bar{\varepsilon} = 10^{-5}$. We find a pair $(d,\ \lambda)$ which satisfies conditions (4.1)–(4.4) by the primal-dual interior point method [20]. It is notable that the parameters ν, $\bar{M}$, t and the Slater point $\bar{x}$ have an essential role in our algorithms. Specifically, the parameters ν and t have some effect on conditions (4.1)–(4.4) and the line searches, the parameter $\bar{M}$ has some effect on the line search and the Slater point $\bar{x}$ has some effect on the form of subproblem (1.3). In order to take into account these effects, we test our algorithms with the parameters $\nu = 0.15,\ 0.30,\ 0.45$, $\bar{M} = 0,\ 2,\ 4,\ 6,\ 8,\ 10$ and $t = 0.1,\ 1.0,\ 10$ (we choose a Slater point later on).

The first problem is s394 (see Section 5 in [11]).

$$\begin{cases}
\min \quad f(x) = \displaystyle\sum_{j=1}^{100} (x_j^4 + x_j^2 - 0.01 j x_j)\\
\text{s.t.} \quad g_1(x) = \displaystyle\sum_{j=1}^{100} x_j^2 - 1 \leq 0.
\end{cases}$$

The optimal value of this problem is $f(x_*) = -4.7990$. To solve this problem, we set the infeasible initial point $x_0 = (5, 5, \cdots, 5)^T$ which is given in [11]. For this problem, the Slater point is chosen by

$$\bar{x}_1 = (0, \cdots, 0)^T, \quad \bar{x}_2 = (0.09, \cdots, 0.09)^T \quad \text{and} \quad \bar{x}_3 = (-0.09, \cdots, -0.09)^T.$$

In this case, we have that

$$g_1(\bar{x}_1) = -1, \quad g_1(\bar{x}_2) = g_1(\bar{x}_3) = -0.19$$

and

$$g_1(\bar{x}_1) < g_1(\bar{x}_2) = g_1(\bar{x}_3) < 0.$$

The second problem is HS100 (see [8] or Section 5 in [11]).

$$\begin{cases} \min & f(x) &=& (x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2 \\ & & & +10x_5^6+7x_6^2+x_7^4-4x_6x_7-10x_6-8x_7 \\ \text{s.t.} & g_1(x) &=& 2x_1^2+3x_2^4+x_3+4x_4^2+5x_5-127\le 0, \\ & g_2(x) &=& 7x_1+3x_2+10x_3^2+x_4-x_5-282\le 0, \\ & g_3(x) &=& 23x_1+x_2^2+6x_6^2-8x_7-196\le 0, \\ & g_4(x) &=& 4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-11x_7\le 0. \end{cases}$$

The optimal value of this problem is $f(x_*) = 680.63$. This problem is not a convex programming because the Hessian matrix of the objective function is not positive semidefinite when $x_7^2 < 2/21$. However, all of constraints are convex functions i.e., the feasible region is a convex set. If the problem has a convex feasible region, we can solve such a problem by our algorithms in practice. To solve this problem, we set the infeasible initial point $x_0 = (5, 5, 5, 5, 5, 5, 5)$ which is given in [11]. For this problem, the Slater point is chosen by

$$\bar{x}_4 = (0,0,0,0,0,0,1)^T \quad \text{and} \quad \bar{x}_5 = (8,0,-2,0,0,6,27)^T.$$

In this case, we have that

$$g(\bar{x}_4) = (-127,-282,-196,-11)^T, \quad g(\bar{x}_5) = (-1,-186,-12,-3)^T$$

and

$$g(\bar{x}_4) < g(\bar{x}_5) < 0.$$

The numerical results are given in Tables 3–17. The outlines of Tables 3–17 are given by the following tables (Tables 1 and 2).

Table 1: Outline of Test problem s394

The number of Table	Parameter t	Slater point $\bar{x}$
3	$t_1 = 0.1$	$\bar{x}_1 = (0,\cdots,0)^T$
4	$t_2 = 1$	$\bar{x}_1 = (0,\cdots,0)^T$
5	$t_3 = 10$	$\bar{x}_1 = (0,\cdots,0)^T$
6	$t_1 = 0.1$	$\bar{x}_2 = (0.09,\cdots,0.09)^T$
7	$t_2 = 1$	$\bar{x}_2 = (0.09,\cdots,0.09)^T$
8	$t_3 = 10$	$\bar{x}_2 = (0.09,\cdots,0.09)^T$
9	$t_1 = 0.1$	$\bar{x}_3 = (-0.09,\cdots,-0.09)^T$
10	$t_2 = 1$	$\bar{x}_3 = (-0.09,\cdots,-0.09)^T$
11	$t_3 = 10$	$\bar{x}_3 = (-0.09,\cdots,-0.09)^T$

Table 2: Outline of Test problem HS100

The number of Table	Parameter t	Slater point $\bar{x}$
12	$t_1 = 0.1$	$\bar{x}_4 = (0,0,0,0,0,0,1)^T$
13	$t_2 = 1$	$\bar{x}_4 = (0,0,0,0,0,0,1)^T$
14	$t_3 = 10$	$\bar{x}_4 = (0,0,0,0,0,0,1)^T$
15	$t_1 = 0.1$	$\bar{x}_5 = (8,0,-2,0,0,6,27)^T$
16	$t_2 = 1$	$\bar{x}_5 = (8,0,-2,0,0,6,27)^T$
17	$t_3 = 10$	$\bar{x}_5 = (8,0,-2,0,0,6,27)^T$

The terms in Tables 3–17 mean the following:

"iteration"	-	the number of iterations.
"search"	-	the number of merit function evaluations.
"unit"	-	the number of accepting the unit step size.
"unit (fre)"	-	the frequency of accepting the unit step size.
"precision"	-	the final value of $\|d\|$.
" - "	-	this symbol means the failure of algorithm.

For columns "unit (fre)" and "precision", we use three significant digits.

First, we review numerical results of Test problem s394. As shown in Tables 3–5, 8 and 11, we can observe that there is no difference among our algorithms. From these numerical results and the fact that subproblem (1.3) easily becomes subproblem (1.2) when $g_i(\bar{x})$ $(i = 1, \cdots, m)$ is sufficiently small, we can guess that Slater point $\bar{x}_1$ and parameter t_3 are adequate for Test problem s394. By contrast, as shown in Tables 6 and 7, we can observe that the nonmonotone line searches reduce the number of merit function evaluations (see column "search") except the cases (Table 6, 7, NISQCQP2, $\nu = 0.15,\ 0.30,\ 0.45$ and $\bar{M} = 2,\ 4,\ 6$). However, we can observe that the nonmonotone line searches conversely deteriorate numerical performance in Tables 9 and 10. Secondly, we review numerical results of Test problem HS100. From Tables 12–17, we can guess that Slater point $\bar{x}_4$ is more suitable than $\bar{x}_5$ and that parameter t_3 is most suitable of the three. Though Slater point $\bar{x}_4$ and parameter t_3 are adequate, there is a slight difference among our algorithms. From Table 14, we can find that the nonmonotone line searches reduce the number of merit function evaluations in all cases. However, as shown in Tables 12, 13 and 15–17, we can observe that the nonmonotone line searches (especially, Algorithm NISQCQP1) deteriorate numerical performance in many cases. In this numerical experiment, there are the cases where we cannot solve Test problem HS100 under tolerance $\varepsilon = 10^{-5}$ (see Table 12, 13 and 17). In addition, we can confirm that Theorem 2 (i) holds except the cases which fail and (ii) holds except the case (Table 17, NISQCQP2, $\nu = 0.45$ and $\bar{M} = 4$) and the cases which fail. When we choose the tolerance $\varepsilon = 10^{-4}$, we can solve Test problem

HS100 in these cases and we can avoid the Maratos effect in the case (Table 17, NISQCQP2, $\nu = 0.45$ and $\bar{M} = 4$). These observations suggest that we need to consider a stopping criterion $\|d_k\| \leq \varepsilon$. In conclusion, our numerical results show that the performance of our algorithms depends on choosing a symmetric positive definite matrix B_k (a parameter t) and a Slater point $\bar{x}$ and setting parameters, especially ν and $\bar{M}$ and that the nonmonotone line search may improve numerical performance a the suitable matrix B_k and Slater point $\bar{x}$.

Table 3: Test problem s394 (t_1, $\bar{x}_1$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		10	10	10	1.000	4.61E-07
NISQCQP1	0.15	2	10	10	10	1.000	4.61E-07
NISQCQP1	0.15	4	10	10	10	1.000	4.61E-07
NISQCQP1	0.15	6	10	10	10	1.000	4.61E-07
NISQCQP1	0.15	8	10	10	10	1.000	4.61E-07
NISQCQP1	0.15	10	10	10	10	1.000	4.61E-07
NISQCQP2	0.15	2	10	10	10	1.000	4.61E-07
NISQCQP2	0.15	4	10	10	10	1.000	4.61E-07
NISQCQP2	0.15	6	10	10	10	1.000	4.61E-07
NISQCQP2	0.15	8	10	10	10	1.000	4.61E-07
NISQCQP2	0.15	10	10	10	10	1.000	4.61E-07
ISQCQP	0.30		10	10	10	1.000	4.02E-07
NISQCQP1	0.30	2	10	10	10	1.000	4.02E-07
NISQCQP1	0.30	4	10	10	10	1.000	4.02E-07
NISQCQP1	0.30	6	10	10	10	1.000	4.02E-07
NISQCQP1	0.30	8	10	10	10	1.000	4.02E-07
NISQCQP1	0.30	10	10	10	10	1.000	4.02E-07
NISQCQP2	0.30	2	10	10	10	1.000	4.02E-07
NISQCQP2	0.30	4	10	10	10	1.000	4.02E-07
NISQCQP2	0.30	6	10	10	10	1.000	4.02E-07
NISQCQP2	0.30	8	10	10	10	1.000	4.02E-07
NISQCQP2	0.30	10	10	10	10	1.000	4.02E-07
ISQCQP	0.45		10	10	10	1.000	4.95E-07
NISQCQP1	0.45	2	10	10	10	1.000	4.95E-07
NISQCQP1	0.45	4	10	10	10	1.000	4.95E-07
NISQCQP1	0.45	6	10	10	10	1.000	4.95E-07
NISQCQP1	0.45	8	10	10	10	1.000	4.95E-07
NISQCQP1	0.45	10	10	10	10	1.000	4.95E-07
NISQCQP2	0.45	2	10	10	10	1.000	4.95E-07
NISQCQP2	0.45	4	10	10	10	1.000	4.95E-07
NISQCQP2	0.45	6	10	10	10	1.000	4.95E-07
NISQCQP2	0.45	8	10	10	10	1.000	4.95E-07
NISQCQP2	0.45	10	10	10	10	1.000	4.95E-07

Table 4: Test problem s394 (t_2, $\bar{x}_1$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		11	11	11	1.000	1.54E-08
NISQCQP1	0.15	2	11	11	11	1.000	1.54E-08
NISQCQP1	0.15	4	11	11	11	1.000	1.54E-08
NISQCQP1	0.15	6	11	11	11	1.000	1.54E-08
NISQCQP1	0.15	8	11	11	11	1.000	1.54E-08
NISQCQP1	0.15	10	11	11	11	1.000	1.54E-08
NISQCQP2	0.15	2	11	11	11	1.000	1.54E-08
NISQCQP2	0.15	4	11	11	11	1.000	1.54E-08
NISQCQP2	0.15	6	11	11	11	1.000	1.54E-08
NISQCQP2	0.15	8	11	11	11	1.000	1.54E-08
NISQCQP2	0.15	10	11	11	11	1.000	1.54E-08
ISQCQP	0.30		11	11	11	1.000	1.54E-08
NISQCQP1	0.30	2	11	11	11	1.000	1.54E-08
NISQCQP1	0.30	4	11	11	11	1.000	1.54E-08
NISQCQP1	0.30	6	11	11	11	1.000	1.54E-08
NISQCQP1	0.30	8	11	11	11	1.000	1.54E-08
NISQCQP1	0.30	10	11	11	11	1.000	1.54E-08
NISQCQP2	0.30	2	11	11	11	1.000	1.54E-08
NISQCQP2	0.30	4	11	11	11	1.000	1.54E-08
NISQCQP2	0.30	6	11	11	11	1.000	1.54E-08
NISQCQP2	0.30	8	11	11	11	1.000	1.54E-08
NISQCQP2	0.30	10	11	11	11	1.000	1.54E-08
ISQCQP	0.45		11	11	11	1.000	1.54E-08
NISQCQP1	0.45	2	11	11	11	1.000	1.54E-08
NISQCQP1	0.45	4	11	11	11	1.000	1.54E-08
NISQCQP1	0.45	6	11	11	11	1.000	1.54E-08
NISQCQP1	0.45	8	11	11	11	1.000	1.54E-08
NISQCQP1	0.45	10	11	11	11	1.000	1.54E-08
NISQCQP2	0.45	2	11	11	11	1.000	1.54E-08
NISQCQP2	0.45	4	11	11	11	1.000	1.54E-08
NISQCQP2	0.45	6	11	11	11	1.000	1.54E-08
NISQCQP2	0.45	8	11	11	11	1.000	1.54E-08
NISQCQP2	0.45	10	11	11	11	1.000	1.54E-08

Table 5: Test problem s394 (t_3, $\bar{x}_1$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		11	11	11	1.000	8.33E-07
NISQCQP1	0.15	2	11	11	11	1.000	8.33E-07
NISQCQP1	0.15	4	11	11	11	1.000	8.33E-07
NISQCQP1	0.15	6	11	11	11	1.000	8.33E-07
NISQCQP1	0.15	8	11	11	11	1.000	8.33E-07
NISQCQP1	0.15	10	11	11	11	1.000	8.33E-07
NISQCQP2	0.15	2	11	11	11	1.000	8.33E-07
NISQCQP2	0.15	4	11	11	11	1.000	8.33E-07
NISQCQP2	0.15	6	11	11	11	1.000	8.33E-07
NISQCQP2	0.15	8	11	11	11	1.000	8.33E-07
NISQCQP2	0.15	10	11	11	11	1.000	8.33E-07
ISQCQP	0.30		11	11	11	1.000	8.33E-07
NISQCQP1	0.30	2	11	11	11	1.000	8.33E-07
NISQCQP1	0.30	4	11	11	11	1.000	8.33E-07
NISQCQP1	0.30	6	11	11	11	1.000	8.33E-07
NISQCQP1	0.30	8	11	11	11	1.000	8.33E-07
NISQCQP1	0.30	10	11	11	11	1.000	8.33E-07
NISQCQP2	0.30	2	11	11	11	1.000	8.33E-07
NISQCQP2	0.30	4	11	11	11	1.000	8.33E-07
NISQCQP2	0.30	6	11	11	11	1.000	8.33E-07
NISQCQP2	0.30	8	11	11	11	1.000	8.33E-07
NISQCQP2	0.30	10	11	11	11	1.000	8.33E-07
ISQCQP	0.45		11	11	11	1.000	8.33E-07
NISQCQP1	0.45	2	11	11	11	1.000	8.33E-07
NISQCQP1	0.45	4	11	11	11	1.000	8.33E-07
NISQCQP1	0.45	6	11	11	11	1.000	8.33E-07
NISQCQP1	0.45	8	11	11	11	1.000	8.33E-07
NISQCQP1	0.45	10	11	11	11	1.000	8.33E-07
NISQCQP2	0.45	2	11	11	11	1.000	8.33E-07
NISQCQP2	0.45	4	11	11	11	1.000	8.33E-07
NISQCQP2	0.45	6	11	11	11	1.000	8.33E-07
NISQCQP2	0.45	8	11	11	11	1.000	8.33E-07
NISQCQP2	0.45	10	11	11	11	1.000	8.33E-07

Table 6: Test problem s394 (t_1, $\bar{x}_2$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		11	15	10	0.909	1.50E-08
NISQCQP1	0.15	2	12	12	12	1.000	7.10E-08
NISQCQP1	0.15	4	12	12	12	1.000	7.10E-08
NISQCQP1	0.15	6	12	12	12	1.000	7.10E-08
NISQCQP1	0.15	8	12	12	12	1.000	7.10E-08
NISQCQP1	0.15	10	12	12	12	1.000	7.10E-08
NISQCQP2	0.15	2	11	15	10	0.909	1.50E-08
NISQCQP2	0.15	4	11	15	10	0.909	1.50E-08
NISQCQP2	0.15	6	12	22	11	0.916	7.10E-08
NISQCQP2	0.15	8	12	12	12	1.000	7.10E-08
NISQCQP2	0.15	10	12	12	12	1.000	7.10E-08
ISQCQP	0.30		11	15	10	0.909	1.40E-08
NISQCQP1	0.30	2	12	12	12	1.000	4.74E-08
NISQCQP1	0.30	4	12	12	12	1.000	4.74E-08
NISQCQP1	0.30	6	12	12	12	1.000	4.74E-08
NISQCQP1	0.30	8	12	12	12	1.000	4.74E-08
NISQCQP1	0.30	10	12	12	12	1.000	4.74E-08
NISQCQP2	0.30	2	11	15	10	0.909	1.40E-08
NISQCQP2	0.30	4	11	15	10	0.909	1.40E-08
NISQCQP2	0.30	6	12	23	11	0.916	3.70E-08
NISQCQP2	0.30	8	12	12	12	1.000	4.74E-08
NISQCQP2	0.30	10	12	12	12	1.000	4.74E-08
ISQCQP	0.45		11	15	10	0.909	1.88E-08
NISQCQP1	0.45	2	12	12	12	1.000	6.09E-08
NISQCQP1	0.45	4	12	12	12	1.000	6.09E-08
NISQCQP1	0.45	6	12	12	12	1.000	6.09E-08
NISQCQP1	0.45	8	12	12	12	1.000	6.09E-08
NISQCQP1	0.45	10	12	12	12	1.000	6.09E-08
NISQCQP2	0.45	2	11	15	10	0.909	1.88E-08
NISQCQP2	0.45	4	11	15	10	0.909	1.88E-08
NISQCQP2	0.45	6	12	24	11	0.916	5.93E-08
NISQCQP2	0.45	8	12	12	12	1.000	6.09E-08
NISQCQP2	0.45	10	12	12	12	1.000	6.09E-08

Table 7: Test problem s394 (t_2, $\bar{x}_2$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		11	14	10	0.909	1.07E-06
NISQCQP1	0.15	2	13	13	13	1.000	5.41E-08
NISQCQP1	0.15	4	13	13	13	1.000	5.41E-08
NISQCQP1	0.15	6	13	13	13	1.000	5.41E-08
NISQCQP1	0.15	8	13	13	13	1.000	5.41E-08
NISQCQP1	0.15	10	13	13	13	1.000	5.41E-08
NISQCQP2	0.15	2	11	14	10	0.909	1.07E-06
NISQCQP2	0.15	4	11	14	10	0.909	1.07E-06
NISQCQP2	0.15	6	13	23	12	0.923	1.54E-08
NISQCQP2	0.15	8	13	13	13	1.000	5.41E-08
NISQCQP2	0.15	10	13	13	13	1.000	5.41E-08
ISQCQP	0.30		11	14	10	0.909	1.07E-06
NISQCQP1	0.30	2	13	13	13	1.000	5.41E-08
NISQCQP1	0.30	4	13	13	13	1.000	5.41E-08
NISQCQP1	0.30	6	13	13	13	1.000	5.41E-08
NISQCQP1	0.30	8	13	13	13	1.000	5.41E-08
NISQCQP1	0.30	10	13	13	13	1.000	5.41E-08
NISQCQP2	0.30	2	11	14	10	0.909	1.07E-06
NISQCQP2	0.30	4	11	14	10	0.909	1.07E-06
NISQCQP2	0.30	6	13	24	12	0.923	1.59E-08
NISQCQP2	0.30	8	13	13	13	1.000	5.41E-08
NISQCQP2	0.30	10	13	13	13	1.000	5.41E-08
ISQCQP	0.45		11	15	10	0.909	2.05E-06
NISQCQP1	0.45	2	13	13	13	1.000	5.41E-08
NISQCQP1	0.45	4	13	13	13	1.000	5.41E-08
NISQCQP1	0.45	6	13	13	13	1.000	5.41E-08
NISQCQP1	0.45	8	13	13	13	1.000	5.41E-08
NISQCQP1	0.45	10	13	13	13	1.000	5.41E-08
NISQCQP2	0.45	2	11	15	10	0.909	2.05E-06
NISQCQP2	0.45	4	11	14	10	0.909	2.05E-06
NISQCQP2	0.45	6	13	25	12	0.923	1.70E-08
NISQCQP2	0.45	8	13	13	13	1.000	5.41E-08
NISQCQP2	0.45	10	13	13	13	1.000	5.41E-08

Table 8: Test problem s394 (t_3, $\bar{x}_2$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		12	12	12	1.000	3.01E-08
NISQCQP1	0.15	2	12	12	12	1.000	3.01E-08
NISQCQP1	0.15	4	12	12	12	1.000	3.01E-08
NISQCQP1	0.15	6	12	12	12	1.000	3.01E-08
NISQCQP1	0.15	8	12	12	12	1.000	3.01E-08
NISQCQP1	0.15	10	12	12	12	1.000	3.01E-08
NISQCQP2	0.15	2	12	12	12	1.000	3.01E-08
NISQCQP2	0.15	4	12	12	12	1.000	3.01E-08
NISQCQP2	0.15	6	12	12	12	1.000	3.01E-08
NISQCQP2	0.15	8	12	12	12	1.000	3.01E-08
NISQCQP2	0.15	10	12	12	12	1.000	3.01E-08
ISQCQP	0.30		12	12	12	1.000	3.01E-08
NISQCQP1	0.30	2	12	12	12	1.000	3.01E-08
NISQCQP1	0.30	4	12	12	12	1.000	3.01E-08
NISQCQP1	0.30	6	12	12	12	1.000	3.01E-08
NISQCQP1	0.30	8	12	12	12	1.000	3.01E-08
NISQCQP1	0.30	10	12	12	12	1.000	3.01E-08
NISQCQP2	0.30	2	12	12	12	1.000	3.01E-08
NISQCQP2	0.30	4	12	12	12	1.000	3.01E-08
NISQCQP2	0.30	6	12	12	12	1.000	3.01E-08
NISQCQP2	0.30	8	12	12	12	1.000	3.01E-08
NISQCQP2	0.30	10	12	12	12	1.000	3.01E-08
ISQCQP	0.45		12	12	12	1.000	3.01E-08
NISQCQP1	0.45	2	12	12	12	1.000	3.01E-08
NISQCQP1	0.45	4	12	12	12	1.000	3.01E-08
NISQCQP1	0.45	6	12	12	12	1.000	3.01E-08
NISQCQP1	0.45	8	12	12	12	1.000	3.01E-08
NISQCQP1	0.45	10	12	12	12	1.000	3.01E-08
NISQCQP2	0.45	2	12	12	12	1.000	3.01E-08
NISQCQP2	0.45	4	12	12	12	1.000	3.01E-08
NISQCQP2	0.45	6	12	12	12	1.000	3.01E-08
NISQCQP2	0.45	8	12	12	12	1.000	3.01E-08
NISQCQP2	0.45	10	12	12	12	1.000	3.01E-08

Table 9: Test problem s394 (t_1, $\bar{x}_3$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		12	20	10	0.833	2.39E-06
NISQCQP1	0.15	2	23	35	20	0.869	2.08E-10
NISQCQP1	0.15	4	80	139	50	0.625	3.27E-10
NISQCQP1	0.15	6	58	79	42	0.724	6.29E-11
NISQCQP1	0.15	8	126	178	89	0.706	1.44E-10
NISQCQP1	0.15	10	98	131	72	0.734	7.93E-11
NISQCQP2	0.15	2	12	20	10	0.833	2.39E-06
NISQCQP2	0.15	4	12	20	10	0.833	2.39E-06
NISQCQP2	0.15	6	17	28	14	0.823	5.99E-07
NISQCQP2	0.15	8	22	35	17	0.772	8.52E-07
NISQCQP2	0.15	10	26	39	21	0.807	8.97E-07
ISQCQP	0.30		12	20	10	0.833	2.41E-06
NISQCQP1	0.30	2	23	35	20	0.869	1.64E-10
NISQCQP1	0.30	4	79	137	53	0.670	3.06E-10
NISQCQP1	0.30	6	199	348	102	0.512	1.70E-10
NISQCQP1	0.30	8	76	99	58	0.763	8.97E-11
NISQCQP1	0.30	10	329	564	180	0.547	2.86E-10
NISQCQP2	0.30	2	12	20	10	0.833	2.41E-06
NISQCQP2	0.30	4	12	20	10	0.833	2.41E-06
NISQCQP2	0.30	6	19	33	14	0.736	3.15E-07
NISQCQP2	0.30	8	22	35	17	0.772	8.09E-07
NISQCQP2	0.30	10	26	39	21	0.807	8.85E-07
ISQCQP	0.45		12	20	10	0.833	2.92E-06
NISQCQP1	0.45	2	27	43	23	0.851	7.24E-11
NISQCQP1	0.45	4	26	35	21	0.807	3.51E-10
NISQCQP1	0.45	6	194	337	110	0.567	3.14E-10
NISQCQP1	0.45	8	81	106	62	0.765	9.28E-11
NISQCQP1	0.45	10	280	472	162	0.578	2.65E-10
NISQCQP2	0.45	2	12	20	10	0.833	2.92E-06
NISQCQP2	0.45	4	12	20	10	0.833	2.92E-06
NISQCQP2	0.45	6	17	28	14	0.823	8.69E-07
NISQCQP2	0.45	8	22	36	17	0.772	2.33E-06
NISQCQP2	0.45	10	26	40	21	0.807	2.66E-06

Table 10: Test problem s394 ($t_2,\ \bar{x}_3$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		13	39	11	0.846	1.05E-08
NISQCQP1	0.15	2	14	17	13	0.928	9.68E-08
NISQCQP1	0.15	4	41	54	33	0.804	6.03E-06
NISQCQP1	0.15	6	77	102	59	0.766	1.40E-07
NISQCQP1	0.15	8	86	111	67	0.779	4.03E-08
NISQCQP1	0.15	10	94	122	73	0.776	2.99E-08
NISQCQP2	0.15	2	13	19	11	0.846	1.05E-08
NISQCQP2	0.15	4	13	19	11	0.846	1.05E-08
NISQCQP2	0.15	6	18	30	14	0.777	8.57E-06
NISQCQP2	0.15	8	21	31	17	0.809	2.92E-09
NISQCQP2	0.15	10	29	39	25	0.862	9.87E-06
ISQCQP	0.30		13	19	11	0.846	1.05E-08
NISQCQP1	0.30	2	14	17	13	0.928	9.68E-08
NISQCQP1	0.30	4	41	54	33	0.804	6.03E-06
NISQCQP1	0.30	6	73	97	56	0.767	1.39E-07
NISQCQP1	0.30	8	84	110	65	0.773	3.79E-08
NISQCQP1	0.30	10	92	121	71	0.771	2.82E-08
NISQCQP2	0.30	2	13	19	11	0.846	1.05E-08
NISQCQP2	0.30	4	13	19	11	0.846	1.05E-08
NISQCQP2	0.30	6	18	30	14	0.777	8.57E-06
NISQCQP2	0.30	8	21	31	17	0.809	2.92E-09
NISQCQP2	0.30	10	25	35	21	0.840	2.96E-09
ISQCQP	0.45		14	26	11	0.785	1.30E-06
NISQCQP1	0.45	2	14	17	13	0.928	9.68E-08
NISQCQP1	0.45	4	33	44	26	0.787	9.96E-06
NISQCQP1	0.45	6	88	123	64	0.727	4.21E-07
NISQCQP1	0.45	8	84	110	65	0.773	3.79E-08
NISQCQP1	0.45	10	88	115	68	0.772	3.81E-08
NISQCQP2	0.45	2	14	26	11	0.785	1.30E-06
NISQCQP2	0.45	4	14	26	11	0.785	1.30E-06
NISQCQP2	0.45	6	16	25	13	0.812	9.36E-06
NISQCQP2	0.45	8	21	31	17	0.809	2.92E-09
NISQCQP2	0.45	10	23	33	19	0.826	9.81E-06

Table 11: Test problem s394 ($t_3,\ \bar{x}_3$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		13	13	13	1.000	7.58E-08
NISQCQP1	0.15	2	13	13	13	1.000	7.58E-08
NISQCQP1	0.15	4	13	13	13	1.000	7.58E-08
NISQCQP1	0.15	6	13	13	13	1.000	7.58E-08
NISQCQP1	0.15	8	13	13	13	1.000	7.58E-08
NISQCQP1	0.15	10	13	13	13	1.000	7.58E-08
NISQCQP2	0.15	2	13	13	13	1.000	7.58E-08
NISQCQP2	0.15	4	13	13	13	1.000	7.58E-08
NISQCQP2	0.15	6	13	13	13	1.000	7.58E-08
NISQCQP2	0.15	8	13	13	13	1.000	7.58E-08
NISQCQP2	0.15	10	13	13	13	1.000	7.58E-08
ISQCQP	0.30		13	13	13	1.000	7.58E-08
NISQCQP1	0.30	2	13	13	13	1.000	7.58E-08
NISQCQP1	0.30	4	13	13	13	1.000	7.58E-08
NISQCQP1	0.30	6	13	13	13	1.000	7.58E-08
NISQCQP1	0.30	8	13	13	13	1.000	7.58E-08
NISQCQP1	0.30	10	13	13	13	1.000	7.58E-08
NISQCQP2	0.30	2	13	13	13	1.000	7.58E-08
NISQCQP2	0.30	4	13	13	13	1.000	7.58E-08
NISQCQP2	0.30	6	13	13	13	1.000	7.58E-08
NISQCQP2	0.30	8	13	13	13	1.000	7.58E-08
NISQCQP2	0.30	10	13	13	13	1.000	7.58E-08
ISQCQP	0.45		13	13	13	1.000	7.58E-08
NISQCQP1	0.45	2	13	13	13	1.000	7.58E-08
NISQCQP1	0.45	4	13	13	13	1.000	7.58E-08
NISQCQP1	0.45	6	13	13	13	1.000	7.58E-08
NISQCQP1	0.45	8	13	13	13	1.000	7.58E-08
NISQCQP1	0.45	10	13	13	13	1.000	7.58E-08
NISQCQP2	0.45	2	13	13	13	1.000	7.58E-08
NISQCQP2	0.45	4	13	13	13	1.000	7.58E-08
NISQCQP2	0.45	6	13	13	13	1.000	7.58E-08
NISQCQP2	0.45	8	13	13	13	1.000	7.58E-08
NISQCQP2	0.45	10	13	13	13	1.000	7.58E-08

Table 12: Test problem HS100 ($t_1,\ \bar{x}_4$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		17	42	14	0.823	1.09E-09
NISQCQP1	0.15	2	43	111	25	0.581	7.47E-07
NISQCQP1	0.15	4	97	249	56	0.577	3.54E-09
NISQCQP1	0.15	6	89	195	63	0.707	2.20E-09
NISQCQP1	0.15	8	213	550	127	0.539	7.44E-07
NISQCQP1	0.15	10	221	603	127	0.574	2.80E-07
NISQCQP2	0.15	2	23	71	14	0.608	1.92E-08
NISQCQP2	0.15	4	19	46	15	0.789	6.47E-08
NISQCQP2	0.15	6	22	60	15	0.681	8.06E-06
NISQCQP2	0.15	8	29	85	20	0.689	3.19E-06
NISQCQP2	0.15	10	37	63	31	0.837	3.59E-06
ISQCQP	0.30		17	42	14	0.823	1.02E-09
NISQCQP1	0.30	2	56	154	27	0.482	6.77E-08
NISQCQP1	0.30	4	81	204	49	0.604	4.36E-08
NISQCQP1	0.30	6	70	134	53	0.757	4.46E-09
NISQCQP1	0.30	8	201	552	101	0.502	1.70E-07
NISQCQP1	0.30	10	249	646	155	0.622	3.00E-06
NISQCQP2	0.30	2	23	71	14	0.608	1.31E-08
NISQCQP2	0.30	4	22	63	16	0.727	1.82E-06
NISQCQP2	0.30	6	-	-	-	-	-
NISQCQP2	0.30	8	30	87	20	0.666	8.16E-06
NISQCQP2	0.30	10	32	56	28	0.875	1.21E-06
ISQCQP	0.45		-	-	-	-	-
NISQCQP1	0.45	2	45	120	25	0.555	2.84E-07
NISQCQP1	0.45	4	86	224	54	0.627	8.85E-07
NISQCQP1	0.45	6	89	169	60	0.674	1.11E-06
NISQCQP1	0.45	8	196	47	119	0.607	1.14E-07
NISQCQP1	0.45	10	42	44	41	0.976	4.32E-08
NISQCQP2	0.45	2	24	75	13	0.541	5.19E-09
NISQCQP2	0.45	4	22	65	16	0.727	3.59E-06
NISQCQP2	0.45	6	20	47	15	0.750	4.20E-07
NISQCQP2	0.45	8	25	62	19	0.760	7.45E-09
NISQCQP2	0.45	10	32	50	27	0.843	2.42E-06

Table 13: Test problem HS100 ($t_2,\ \bar{x}_4$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		17	43	13	0.764	6.48E-06
NISQCQP1	0.15	2	37	87	26	0.702	2.53E-07
NISQCQP1	0.15	4	60	136	38	0.633	3.06E-06
NISQCQP1	0.15	6	75	165	49	0.653	3.46E-06
NISQCQP1	0.15	8	105	232	69	0.657	2.49E-06
NISQCQP1	0.15	10	152	299	103	0.677	2.49E-06
NISQCQP2	0.15	2	20	44	15	0.750	9.78E-07
NISQCQP2	0.15	4	20	44	15	0.750	9.78E-07
NISQCQP2	0.15	6	22	43	18	0.818	1.21E-06
NISQCQP2	0.15	8	-	-	-	-	-
NISQCQP2	0.15	10	25	52	19	0.760	4.46E-06
ISQCQP	0.30		17	43	13	0.764	7.15E-06
NISQCQP1	0.30	2	35	80	25	0.714	2.63E-07
NISQCQP1	0.30	4	58	131	37	0.637	2.97E-06
NISQCQP1	0.30	6	75	165	49	0.653	3.45E-06
NISQCQP1	0.30	8	107	230	71	0.663	2.49E-06
NISQCQP1	0.30	10	136	276	91	0.669	2.48E-06
NISQCQP2	0.30	2	18	32	15	0.833	9.37E-08
NISQCQP2	0.30	4	20	44	15	0.750	9.78E-07
NISQCQP2	0.30	6	22	48	16	0.727	5.62E-06
NISQCQP2	0.30	8	-	-	-	-	-
NISQCQP2	0.30	10	25	52	19	0.760	4.46E-06
ISQCQP	0.45		17	42	13	0.764	3.10E-07
NISQCQP1	0.45	2	31	56	23	0.741	1.84E-08
NISQCQP1	0.45	4	53	113	36	0.679	3.70E-06
NISQCQP1	0.45	6	67	145	46	0.686	3.18E-06
NISQCQP1	0.45	8	44	54	38	0.863	2.71E-09
NISQCQP1	0.45	10	64	81	52	0.812	3.01E-07
NISQCQP2	0.45	2	16	30	13	0.812	2.09E-06
NISQCQP2	0.45	4	18	37	14	0.777	1.57E-06
NISQCQP2	0.45	6	19	37	17	0.894	2.23E-06
NISQCQP2	0.45	8	24	55	19	0.791	3.35E-07
NISQCQP2	0.45	10	36	62	28	0.777	2.31E-07

Table 14: Test problem HS100 (t_3, $\bar{x}_4$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		18	27	15	0.833	2.89E-06
NISQCQP1	0.15	2	18	18	18	1.000	2.83E-08
NISQCQP1	0.15	4	18	18	18	1.000	2.83E-08
NISQCQP1	0.15	6	18	18	18	1.000	2.83E-08
NISQCQP1	0.15	8	18	18	18	1.000	2.83E-08
NISQCQP1	0.15	10	18	18	18	1.000	2.83E-08
NISQCQP2	0.15	2	18	25	16	0.888	4.32E-07
NISQCQP2	0.15	4	18	23	16	0.888	1.21E-06
NISQCQP2	0.15	6	18	18	18	1.000	2.83E-08
NISQCQP2	0.15	8	18	18	18	1.000	2.83E-08
NISQCQP2	0.15	10	18	18	18	1.000	2.83E-08
ISQCQP	0.30		17	27	14	0.823	5.65E-06
NISQCQP1	0.30	2	18	18	18	1.000	2.83E-08
NISQCQP1	0.30	4	18	18	18	1.000	2.83E-08
NISQCQP1	0.30	6	18	18	18	1.000	2.83E-08
NISQCQP1	0.30	8	18	18	18	1.000	2.83E-08
NISQCQP1	0.30	10	18	18	18	1.000	2.83E-08
NISQCQP2	0.30	2	18	25	16	0.888	4.32E-07
NISQCQP2	0.30	4	18	23	16	0.888	1.21E-06
NISQCQP2	0.30	6	18	18	18	1.000	2.83E-08
NISQCQP2	0.30	8	18	18	18	1.000	2.83E-08
NISQCQP2	0.30	10	18	18	18	1.000	2.83E-08
ISQCQP	0.45		17	27	14	0.823	1.01E-06
NISQCQP1	0.45	2	18	18	18	1.000	3.14E-08
NISQCQP1	0.45	4	18	18	18	1.000	3.14E-08
NISQCQP1	0.45	6	18	18	18	1.000	3.14E-08
NISQCQP1	0.45	8	18	18	18	1.000	3.14E-08
NISQCQP1	0.45	10	18	18	18	1.000	3.14E-08
NISQCQP2	0.45	2	18	25	16	0.888	4.50E-07
NISQCQP2	0.45	4	18	23	16	0.888	1.33E-06
NISQCQP2	0.45	6	18	18	18	1.000	3.14E-08
NISQCQP2	0.45	8	18	18	18	1.000	3.14E-08
NISQCQP2	0.45	10	18	18	18	1.000	3.14E-08

Table 15: Test problem HS100 (t_1, $\bar{x}_5$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		29	130	12	0.413	4.92E-06
NISQCQP1	0.15	2	172	627	69	0.401	1.72E-09
NISQCQP1	0.15	4	308	1061	116	0.376	5.05E-10
NISQCQP1	0.15	6	834	2953	332	0.398	1.42E-09
NISQCQP1	0.15	8	804	2803	315	0.391	1.52E-10
NISQCQP1	0.15	10	1171	3971	464	0.396	3.47E-10
NISQCQP2	0.15	2	34	138	13	0.382	3.01E-07
NISQCQP2	0.15	4	36	146	17	0.472	3.44E-07
NISQCQP2	0.15	6	29	105	16	0.551	2.28E-07
NISQCQP2	0.15	8	36	128	21	0.583	4.81E-08
NISQCQP2	0.15	10	42	143	22	0.523	3.31E-07
ISQCQP	0.30		59	319	23	0.389	4.04E-07
NISQCQP1	0.30	2	64	209	33	0.515	2.82E-09
NISQCQP1	0.30	4	471	1682	185	0.392	8.56E-10
NISQCQP1	0.30	6	319	1018	139	0.435	1.39E-09
NISQCQP1	0.30	8	991	3524	375	0.378	1.59E-09
NISQCQP1	0.30	10	777	2552	336	0.432	3.87E-11
NISQCQP2	0.30	2	27	111	14	0.518	2.82E-06
NISQCQP2	0.30	4	33	129	17	0.515	3.45E-08
NISQCQP2	0.30	6	36	137	18	0.500	2.37E-08
NISQCQP2	0.30	8	44	153	20	0.454	3.48E-07
NISQCQP2	0.30	10	37	127	22	0.594	1.13E-06
ISQCQP	0.45		56	300	15	0.267	3.65E-07
NISQCQP1	0.45	2	142	551	58	0.408	1.56E-09
NISQCQP1	0.45	4	259	885	112	0.432	1.64E-11
NISQCQP1	0.45	6	600	2106	245	0.408	2.82E-09
NISQCQP1	0.45	8	820	2847	317	0.386	1.42E-09
NISQCQP1	0.45	10	709	2349	298	0.420	6.09E-10
NISQCQP2	0.45	2	49	257	22	0.448	1.18E-06
NISQCQP2	0.45	4	56	295	24	0.428	1.13E-06
NISQCQP2	0.45	6	57	289	24	0.421	7.71E-07
NISQCQP2	0.45	8	37	144	22	0.594	3.40E-07
NISQCQP2	0.45	10	39	150	22	0.564	4.70E-07

Table 16: Test problem HS100 (t_2, $\bar{x}_5$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		33	147	14	0.424	1.08E-07
NISQCQP1	0.15	2	151	513	68	0.450	5.96E-10
NISQCQP1	0.15	4	488	1668	182	0.372	4.36E-10
NISQCQP1	0.15	6	691	2338	280	0.405	2.96E-10
NISQCQP1	0.15	8	465	1423	211	0.453	7.04E-11
NISQCQP1	0.15	10	714	2098	332	0.464	1.16E-06
NISQCQP2	0.15	2	36	141	19	0.527	5.12E-08
NISQCQP2	0.15	4	36	141	19	0.527	5.12E-08
NISQCQP2	0.15	6	41	161	22	0.536	1.14E-07
NISQCQP2	0.15	8	38	141	22	0.578	1.18E-07
NISQCQP2	0.15	10	35	115	22	0.628	3.08E-07
ISQCQP	0.30		87	432	10	0.114	1.87E-07
NISQCQP1	0.30	2	125	414	56	0.448	6.72E-10
NISQCQP1	0.30	4	432	1513	168	0.388	3.98E-10
NISQCQP1	0.30	6	608	2030	254	0.417	1.85E-10
NISQCQP1	0.30	8	948	3197	385	0.406	2.47E-10
NISQCQP1	0.30	10	1215	4050	503	0.413	2.88E-09
NISQCQP2	0.30	2	39	165	18	0.461	1.18E-07
NISQCQP2	0.30	4	35	136	19	0.542	1.11E-07
NISQCQP2	0.30	6	32	111	19	0.593	3.84E-07
NISQCQP2	0.30	8	38	141	22	0.578	1.18E-07
NISQCQP2	0.30	10	32	98	21	0.656	4.32E-06
ISQCQP	0.45		40	194	14	0.350	2.05E-06
NISQCQP1	0.45	2	146	536	58	0.397	7.36E-11
NISQCQP1	0.45	4	368	1259	145	0.394	5.87E-10
NISQCQP1	0.45	6	510	1758	196	0.384	7.71E-10
NISQCQP1	0.45	8	861	2953	333	0.386	3.01E-09
NISQCQP1	0.45	10	1114	3766	432	0.387	2.61E-09
NISQCQP2	0.45	2	27	104	15	0.555	3.17E-07
NISQCQP2	0.45	4	24	76	15	0.625	2.44E-07
NISQCQP2	0.45	6	31	118	20	0.645	1.42E-06
NISQCQP2	0.45	8	40	166	23	0.575	1.46E-07
NISQCQP2	0.45	10	57	198	33	0.578	1.60E-07

Table 17: Test problem HS100 (t_3, $\bar{x}_5$)

Algorithm	ν	$\bar{M}$	iteration	search	unit	unit (fre)	precision
ISQCQP	0.15		33	124	17	0.515	4.92E-06
NISQCQP1	0.15	2	55	124	33	0.600	2.47E-06
NISQCQP1	0.15	4	149	329	86	0.577	3.18E-08
NISQCQP1	0.15	6	92	146	64	0.695	2.75E-08
NISQCQP1	0.15	8	388	876	218	0.561	2.70E-08
NISQCQP1	0.15	10	165	258	112	0.678	4.33E-06
NISQCQP2	0.15	2	28	81	16	0.571	7.92E-06
NISQCQP2	0.15	4	24	56	16	0.666	9.90E-06
NISQCQP2	0.15	6	41	121	25	0.609	1.19E-06
NISQCQP2	0.15	8	33	68	25	0.757	9.73E-08
NISQCQP2	0.15	10	35	69	26	0.742	1.29E-07
ISQCQP	0.30		-	-	-	-	-
NISQCQP1	0.30	2	55	124	33	0.600	2.48E-06
NISQCQP1	0.30	4	180	482	90	0.500	4.95E-08
NISQCQP1	0.30	6	237	541	128	0.540	5.84E-09
NISQCQP1	0.30	8	324	735	175	0.540	3.54E-08
NISQCQP1	0.30	10	381	863	199	0.522	2.21E-08
NISQCQP2	0.30	2	28	81	16	0.571	7.94E-06
NISQCQP2	0.30	4	25	57	17	0.680	3.01E-08
NISQCQP2	0.30	6	-	-	-	-	-
NISQCQP2	0.30	8	31	49	24	0.774	7.85E-09
NISQCQP2	0.30	10	35	69	26	0.742	1.72E-07
ISQCQP	0.45		30	106	15	0.500	6.24E-06
NISQCQP1	0.45	2	116	339	54	0.465	4.63E-08
NISQCQP1	0.45	4	287	930	120	0.418	3.22E-08
NISQCQP1	0.45	6	316	833	165	0.522	3.53E-06
NISQCQP1	0.45	8	241	515	145	0.601	2.70E-08
NISQCQP1	0.45	10	316	713	170	0.537	5.83E-09
NISQCQP2	0.45	2	30	100	15	0.500	3.72E-06
NISQCQP2	0.45	4	54	763	18	0.333	9.92E-06
NISQCQP2	0.45	6	35	98	22	0.628	2.71E-06
NISQCQP2	0.45	8	33	82	23	0.696	7.18E-06
NISQCQP2	0.45	10	38	88	25	0.657	7.98E-06

§5. Concluding remarks

In this paper, we have proposed the inexact SQCQP methods with the nonmonotone line searches and we have shown the global and superlinear convergence properties of our methods. In our numerical experiments, we have been able to confirm that the nonmonotone line search may improve numerical performance under the adequate matrix B_k and Slater point $\bar{x}$. As further works, we would like to propose a feasible and inexact SQCQP method for nonconvex problem and an SQCQP method for a large scale convex programming. Specifically, we plan to construct a sparse approximate symmetric positive definite matrix $G_{k,i}$ instead of the Hessian matrix $\nabla^2 g_i(x_k)$ for $i = 1, \cdots, m$.

Acknowledgments. I am deeply grateful to the referees who gave me the valuable comments.

References

[1] F. Alizadeh and D. Goldfarb, Second-order cone programming, *Mathematical Programming.* **95** (2003), 3–51.

[2] J.F. Bonnans, E.R. Panier, A.L. Tits and J.L. Zhou, Avoiding the Maratos effect by means of a nonmonotone line search II. Inequality constrained problems feasible iterates, *SIAM Journal on Numerical Analysis.* **29** (1992), 1187–1202.

[3] Y.H. Dai, On the nonmonotone line search, *Journal of Optimization Theory and Applications.* **112** (2002), 139–145.

[4] Y.H. Dai and K. Schittkowski, A sequential quadratic programming algorithm with non-monotone line search, *Pacific Journal of Optimization.* **4** (2008), 335–351.

[5] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis.* **23** (1986), 707–716.

[6] M. Fukushima, A successive quadratic programming algorithm with global and superlinear convergence properties, *Mathematical Programming.* **35** (1986), 253–264.

[7] M. Fukushima, Z.Q. Luo and P. Tseng, A sequential quadratically constrained quadratic programming methods for differentiable convex minimization, *SIAM Journal on Optimization.* **13** (2003), 1098–1119.

[8] W. Hock and K. Schittkowski, Test Examples for Nonlinear Programming Codes, *Lecture Notes in Economics and Mathematical Systems*, Vol. 187. Springer, 1981.

[9] J.B. Jian, New sequential quadratically-constrained quadratic programming method of feasible directions and its convergence rate, *Journal of Optimization Theory and Applications.* **129** (2006), 109–130.

[10] J.B. Jian, Q.J. Hu, C.M. Tang and H.Y. Zheng, A sequential quadratically constrained quadratic programming method of feasible directions, *Applied Mathematics and Optimization.* **56** (2007), 343–363.

[11] J.B. Jian, C.M. Tang and H.Y. Zheng, Sequential quadratically constrained quadratic programming norm-relaxed algorithm of strongly sub-feasible directions, *European Journal of Operational Research.* **200** (2010), 645–657.

[12] A. Kato, Y. Narushima and H. Yabe, Global and superlinear convergence of inexact sequential quadratically constrained quadratic programming method for convex programming, *Pacific Journal of Optimization.* **8** (2012), 609–629.

[13] S. Kruk and H. Wolkowicz, SQ2P, sequential quadratic constrained quadratic programming - from *Advances in Nonlinear Programming*, Kluwer Academic, Dordrecht, 1998.

[14] S. Kruk and H. Wolkowicz, Sequential, quadratic constrained, quadratic programming for general nonlinear programming - from *Handbook of Semidefinite Programming*, Kluwer Academic, Dordrecht, 2000.

[15] E.R. Panier and A.L. Tits, Avoiding the Maratos effect by means of a nonmonotone line search I. general constrained problems, *SIAM Journal on Numerical Analysis.* **28** (1991), 1183–1195.

[16] K. Schittkowski, More Test Examples for Nonlinear Programming codes, *Lecture Notes in Economics and Mathematical Systems*, Vol. 282. Spring-Verlag, 1987.

[17] M.V. Solodov, On the sequential quadratically constrained quadratic programming methods, *Mathematics of Operations Research.* **29** (2004), 64–79.

[18] C.M. Tang and J.B. Jian, A sequential quadratically constrained quadratic programming method with an augmented Lagrangian line search function, *Journal of Computational and Applied Mathematics.* **220** (2008), 525–547.

[19] H. Yamashita and H. Yabe, A nonmonotone SQP method with global and superlinear convergence properties, *The Institute of Statistical Mathematics Cooperative Research Report.* **84** (1996), 10–29.

[20] H. Yamashita and H. Yabe, An interior point method with a primal-dual quadratic barrier penalty function for nonlinear optimization, *SIAM Journal on Optimization.* **14** (2003), 479–499.

Atsushi Kato
Department of Mathematical Information Science,
Faculty of Science, Tokyo University of Science
1-3, Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan