AN ADAPTIVELY REGULARIZED SEQUENTIAL QUADRATIC PROGRAMMING METHOD FOR EQUALITY CONSTRAINED OPTIMIZATION

SONGQIANG QIU*
School of Mathematics
China University of Mining and Technology
Xuzhou, Jiangsu, China

ZHONGWEN CHEN
School of Mathematical Sciences
Soochow University
Suzhou, Jiangsu, China

(Communicated by Hoang Xuan Phu)

Abstract. In this paper, we devise an adaptively regularized SQP method for equality constrained optimization problem that is resilient to constraint degeneracy, with a relatively small departure from classical SQP method. The main feature of our method is an adaptively choice of regularization parameter, embedded in a trust-funnel-like algorithmic scheme. Unlike general regularized methods, which update regularization parameter after a regularized problem is approximately solved, our method updates the regularization parameter at each iteration according to the infeasibility measure and the promised improvements achieved by the trial step. The sequence of regularization parameters is not necessarily monotonically decreasing. The whole algorithm is globalized by a trust-funnel-like strategy, in which neither a penalty function nor a filter is needed. We present global and fast local convergence under weak assumptions. Preliminary numerical results on a collection of degenerate problems are reported, which are encouraging.

1. Introduction. We are concerned in this paper with the solution of the equality constrained optimization problem

$$\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad c(x) = 0,
\end{align*}$$

where $x$ is a vector of dimension $n$ and $f(x) : \mathbb{R}^n \to \mathbb{R}$, $c(x) : \mathbb{R}^n \to \mathbb{R}^m$ are assumed to be smooth. The Lagrangian for (1) is defined as

$$\mathcal{L}(x, y) = f(x) + y^T c(x).$$

2020 Mathematics Subject Classification. Primary: 65K05, 90C30, 90C55.
Key words and phrases. Equality constrained optimization, adaptive regularization, trust-funnel-like methods, global convergence, local convergence.

The first author is supported by the Fundamental Research Funds for the Central Universities 2014QNA62.

* Corresponding author: Songqiang Qiu.
The KKT conditions for (1) are
\[
\begin{pmatrix}
\nabla f(x) + \nabla c(x)y \\

\nabla c(x)
\end{pmatrix} =
\begin{pmatrix}
0 \\
0
\end{pmatrix}.
\]

Sequential quadratic programming (SQP) methods [4, 25] are among the most successful methods for the solution of (1). They compute steps via a sequence of quadratic programs (QP):
\[
\min \nabla f_k^T d + \frac{1}{2} d^T H_k d
\]
\[
s.t. \ c_k + \nabla c_k^T d = 0,
\]
where \(\nabla f_k := \nabla f(x_k), \ c_k := c(x_k), \ \nabla c_k := \nabla c(x_k)\) and \(H_k\) is the Hessian (or an approximation of the Hessian) matrix of \(L(x_k, y_k)\). SQP methods for (1) may be interpreted as applying Newton’s method to (2). A Newton-like step \((d_k, y_k^+\) for (2) from \((x_k, y_k)\) solves the linear system
\[
\begin{pmatrix}
H_k & \nabla c_k \\
\nabla c_k^T & 0
\end{pmatrix}
\begin{pmatrix}
d \\
y_k^+ - y_k
\end{pmatrix} =
\begin{pmatrix}
-\nabla L_k \\
c_k
\end{pmatrix}.
\]

As is common for most SQP methods, it should be ensured that \(H_k\) is positive definite on the null space of \(\nabla c_k^T\). If \(\nabla c_k\) is not of full rank, the coefficient matrix
\[
K_0^k = \begin{pmatrix}
H_k & \nabla c_k \\
\nabla c_k^T & 0
\end{pmatrix}
\]
is singular, and a solution of (3) might not exist. Therefore, stabilized SQP (sSQP) or regularization techniques are necessary.

Stabilized SQP (sSQP) methods [24, 29, 23, 56, 33, 30, 14] are specifically designed to remedy the numerical and theoretical performance on degenerate problems. Given the current primal-dual iterate \((x_k, y_k)\) and the stabilization parameter \(v_k > 0\), the sSQP for problem (1) generates a direction of change via the following QP subproblem [33]
\[
\min \nabla f_k^T d + \frac{1}{2} d^T H_k d + \frac{v_k}{2} \|y_k^+\|^2
\]
\[
s.t. \ c_k + \nabla c_k^T d - v_k(y_k^+ - y_k) = 0.
\]
Problem (4) has its own advantage that it is always consistent. Most of existing sSQP methods focus on the local properties of the methods. Few globalization of sSQP methods have been proposed so far. In [24] and [23], sSQP is globalized by combining with a certain primal-dual augmented Lagrangian algorithm. In [14], sSQP is combined with the inexact restoration method. In [30], sSQP is combined with the usual augmented Lagrangian algorithm when dealing with problems having inequality constraints.

In parallel with the development of stabilized SQP methods, regularized methods have been proposed to solve an ill-posed problem by constructing a related problem whose solution is unique and differs only slightly from a solution of the original problem (see, e.g., [22, 45, 35, 20, 25]). One of the regularization schemes is to modify the coefficient matrix by
\[
K_k(\delta_k, v_k) = \begin{pmatrix}
H_k + \delta_k I & \nabla c_k \\
\nabla c_k^T & -v_k I
\end{pmatrix}
\]
for some regularization parameters $\delta_k > 0$ and $v_k > 0$. Consequently, the search direction is obtained by solving

$$
(H_k + \delta_k I) \begin{pmatrix}
\nabla c_k \\
\nabla c_k^T \\
-v_k I
\end{pmatrix}
\begin{pmatrix}
d \\
y_k^+ - y_k
\end{pmatrix} = -\begin{pmatrix}
\nabla \mathcal{L}_k \\
c_k
\end{pmatrix}.
$$

(5)

The connection between these two schemes is easy to verify. The KKT systems for (4) can be written in the form of

$$
(H_k \begin{pmatrix}
\nabla c_k \\
\nabla c_k^T \\
-v_k I
\end{pmatrix}
\begin{pmatrix}
d \\
y_k^+ - y_k
\end{pmatrix} = -\begin{pmatrix}
\nabla \mathcal{L}_k \\
c_k
\end{pmatrix},
$$

which is coincident with (5) in the case where $\delta_k = 0$.

Convergence from arbitrary starting points is always enforced by penalty or penalty-free methods. Both classes of methods carefully keep a balance between reducing objective function and satisfying the constraints. Penalty methods use a merit function, which defines some compromise between reducing the objective function and satisfying the constraints, to keep this balance. Penalty-free methods usually keep this balance by using a filter (see [18, 17, 16, 55, 49] and etc.), trust-funnel-like techniques [27, 42, 43] or other techniques [48, 34, 3, 57].

In this paper, we propose an adaptively regularized SQP method for (1). The presented method embeds the regularization in a trust-funnel-like globalization scheme. Trust funnel method is proposed by Gould and Toint [27]. This method has a nice feature that it does not use any penalty function or a filter. Instead, it is globalized by the trust funnel, which is defined as a progressively decreasing limit on the allowed infeasibility of the iterates. The prediction of the behavior of the algorithm is important in finding an acceptable step. If the objective function is more likely to be reduced, then a step is required to provide sufficient reduction on objective function. Otherwise, significant improvement on feasibility is expected.

In our algorithm, the trust-funnel-like framework is also used in setting regularization parameters. At each iteration, a regularization parameter is determined during the course of finding search direction. If the direction $d_k$ does not promise sufficient reduction on the objective function (as a result, the algorithm focuses on reducing infeasibility), the magnitude of the regularized term $v_k y_k^+$ should be a fraction of the current infeasibility measure. On the other hand, if $d_k$ provides sufficient reduction on the linearize objective function, then $v_k \|y_k^+\|$ should be a fraction of the limit on permitted infeasibility.

Note that the regularization is determined iteratively and may require repeat subproblem solving per step computation. Repeat subproblem solving is common in nonlinear programming solvers such as, for example, trust region methods [8]. In addition, as we will show in the Lemma 3.11, a good estimate of $v_k$, which leads to step admitting above requirements, can be made if the linearized constraints is consistent, which implies that repeat subproblem solving can be avoid in this case.

We use $\|\cdot\|$ to denote the Euclidean norm $\|\cdot\|_2$. Subscript $k$ refers to iteration indices and superscript $(i)$ is the $i$th component of a vector. We define the inertia of a symmetric matrix $P$, which is denoted with $\text{In}(P)$, to be the scalar triple that indicates the numbers $n_+, n_-$, and $n_0$ of positive, negative, and zero eigenvalues, respectively, that is,

$$
\text{In}(P) = (n_+, n_-, n_0).
$$

For a symmetric matrix $G$, we use $\lambda_{\text{min}}^G$ and $\lambda_{\text{max}}^G$ to denote the smallest and largest eigenvalues of $G$. Similarly, we use $\sigma_{\text{max}}^A$ to denote the largest singular value of a matrix $A$. 

The rest of this paper is organized as follows. In Section 2, we present the algorithm. In Section 3, we discuss its global convergence properties. In Section 4, a modified algorithm, which is fast locally convergent under certain assumptions, is given. In Section 5, we give some details of numerical implementation and report preliminary numerical results on DEGEN [10], a set of degenerate test problems. Finally, in Section 6, we give some comments on further work.

2. Algorithm description.

2.1. Regularization and computing step. In short, our algorithm is a trust-funnel-like SQP method with an adaptively regularization scheme. We use a variant of the regularized linear system (5), which is

\[
\begin{pmatrix}
H_k + \delta_k I & \nabla c_k \\
\nabla c_k^T & -v_k I
\end{pmatrix}
\begin{pmatrix}
d \\
y_k^+
\end{pmatrix}
\right) =
\begin{pmatrix}
-\nabla f_k \\
-\nabla c_k
\end{pmatrix}.
\]

At each iteration, the regularization parameter \(v_k\) is initialized as

\[
v_k^{\text{in}} := \min \left\{ \max \{\bar{\nu}, \gamma_v h_k\}, \bar{\nu} \right\},
\]

where \(\bar{\nu} > 0\), \(\gamma_v > 0\) and

\[h_k = \|c_k\|\]

acts as a measure of the infeasibility. Trust-funnel-like algorithm sets an upper limit \(h_k^{\text{max}}\) on permitted infeasibility of the \(k\)–th iterate, i.e., there always holds

\[h_k \leq h_k^{\text{max}}.\]

Suppose that \((d_k, y_k^+)\) solves (6). If \(d_k\) promises sufficient reduction on the linearized objective function in the sense of

\[-\nabla f_k^T d_k \geq \gamma_1 h_k^{\sigma_1},\]

where, \(\gamma_1 > 0\), then the parameter \(v_k\) is required to satisfy

\[v_k \|y_k^+\| \leq \gamma_2 h_k^{\text{max}},\]

where \(\gamma_2 \in (0, 1)\). If (8) is not satisfied, then \(v_k\) should satisfy

\[v_k \|y_k^+\| \leq \gamma_2 h_k.\]

Conditions (9) and (10) ensure that the requirements on feasibility can be met via a line search along \(d_k\). However, a desired regularization parameter is not necessarily available when the current feasibility is bad. To prevent the algorithm from reducing \(v_k\) infinite times or getting a too small regularization parameter, we set a smallest value for \(v_k\). In practice, we set

\[v_k^{\text{min}} = \min \{v_{\text{min}}, \gamma_v^{\text{min}} h_k^{\sigma_v}\},\]

where \(v_{\text{min}} \in (0, 1)\), \(\gamma_v^{\text{min}} > 0\) and \(\sigma_v > 1\). If \(v_k\) becomes smaller than \(v_k^{\text{min}}\), the algorithm resorts to some feasibility restoration algorithm, which tries to get closer to the feasible region. We will describe the details of restoration phase in Section 2.3.

Details of updating \(v_k\) and computing steps are given in Algorithm 1.
Algorithm 1 Regularization and Computing steps

Set $v_k = v_k^{\text{in}}$ where $v_k^{\text{in}}$ is defined as (7). Compute the (approximate) Hessian $H_k$. Set $f_v = 1$.

\begin{algorithmic}
  \State \textbf{if} $\ln(K_k(\delta_k, v_k)) \neq (n, m, 0)$, \textbf{then}
  \State choose $\delta_k > 0$ such that $\ln(K_k(\delta_k, v_k)) = (n, m, 0)$, and set $v = v/2$.
  \State \textbf{end if}
  \State \textbf{repeat}
  \State Solve the linear system (6) to obtain the entry $(d_k, y_k^T)$.
  \State \textbf{if} $(d_k, y_k^T)$ satisfies (8) and (9) \textbf{then}
  \State \hspace{1em} Set $\text{LS}_k = 1$ and $f_v = 0$;
  \State \textbf{else if} $(d_k, y_k^T)$ does not satisfy (8), but satisfies (10) \textbf{then}
  \State \hspace{1em} Set $\text{LS}_k = 2$ and $f_v = 0$;
  \State \textbf{else}
  \State \hspace{1em} if $v_k \geq v_k^{\text{min}}$ \textbf{then}
  \State \hspace{2em} Let $v_k := \frac{v_k}{2}$.
  \State \hspace{1em} \textbf{else}
  \State \hspace{2em} Update $h_k^{\text{max}}$ via (17). Enter the restoration phase (Algorithm 2).
  \State \textbf{end if}
  \State \textbf{end if}
  \State \textbf{end if}
  \State \textbf{until} $f_v = 0$
\end{algorithmic}

2.2. A line search trust-funnel-like framework. Our algorithm allows some flexibility in the line search procedure. In the case where (8) holds, step $d_k$ predicts sufficient reduction on linearized objective function. Hence, sufficient actual reduction on $f(x)$ along this step is expected. For this purpose, a classical Armijo line search is used to find a step size $\alpha_k$ which satisfies

$$f_k - f(x_k + \alpha_k d_k) \geq -\rho \alpha \nabla f_k^T d_k,$$

where $\rho \in (0, 1)$. In this case, we require the infeasibility measure of the next iterate not to exceed $h_k^{\text{max}}$, i.e.,

$$h(x_k + \alpha_k d_k) \leq h_k^{\text{max}}.$$ 

Iterations in this case are called $f$-iterations. If (8) does not hold, which always implies that the current point is far from feasible region, then the algorithm aims to find a step size $\alpha_k$ such that

$$h(x_k + \alpha_k d_k) \leq (1 - \rho \alpha) h_k.$$ 

Iterations in this case are called $h$-iterations.

The update rule of $h_k^{\text{max}}$ is important in driving iterates to the feasible region. At $f$-iterations, $h_k^{\text{max}}$ does not change. At $h$-iterations, $h_k^{\text{max}}$ is updated by

$$h_{k+1}^{\text{max}} = \max \{ \kappa_h h_k^{\text{max}}, \bar{\kappa}_h h_k + (1 - \bar{\kappa}_h) h_{k+1} \},$$

where $\kappa_h$ and $\bar{\kappa}_h$ are constants in $(0, 1)$.

Sometimes, line search procedure may fail to find an acceptable step size, or find a step which is too small to produce sufficient departure from the current point. In these cases, we quit the algorithm temporarily and enter a restoration phase in the hope of finding a new point with better properties by reducing infeasibility. In
practice, we set a lower bound \( \alpha_k^{\text{min}} \) of step size, which is defined by

\[
\alpha_k^{\text{min}} := \begin{cases} 
\min \left\{ \alpha, \gamma \alpha h_k^{\sigma} \min \left\{ \left( \frac{(1 - \rho) \nabla f_d^T d_k}{\|d_k\|^2 + \gamma_p \|d_k\|^2} \right), \left( \frac{(1 - \gamma_2) h_k^{\sigma}}{\|d_k\|^2 + \gamma_p} \right) \right\} \right. & \text{if (8) holds,} \\
\min \left\{ \alpha, \gamma \alpha \left( \frac{(1 - \rho)(1 - \gamma_2)}{\|d_k\|^2 + \gamma_p} \right) \right\} & \text{otherwise,}
\end{cases}
\]

(16)

where \( \alpha \) is a small positive constant, \( \gamma \in (0, \frac{1}{2}) \), \( \sigma > 1 \), \( \sigma > 0 \) and \( \gamma_p \) is a small positive constant used to keep the denominators in (16) bounded away from 0. If \( \alpha_k \) becomes smaller than \( \alpha_k^{\text{min}} \), then the algorithm turns to the restoration phase.

2.3. Feasibility restoration. Restoration phase is necessary for filter methods and this idea applies to our method too. In restoration phase, we first reduce \( h_k^{\text{max}} \) by

\[
h_k^{\text{max}} = \kappa_r h_k^{\text{max}}
\]

(17)

with \( \kappa_r \) a constant in \((0, 1)\). Then, we try to find an iterate \( x_k^r \) satisfying

\[ h(x_k^r) \leq h_k^{\text{max}} \]

by means of iteratively solving the following problem

\[
\text{min } \frac{1}{2} \| c(x) \|_2.
\]

(18)

Given the current iterate \( x_j \) in a restoration course. A direction \( d_j \) is obtained by solving a Levenberg-Marquardt subproblem [32, 38]

\[
\text{min } \frac{1}{2} \| c_j + \nabla c_j^T d \|^2 + \frac{1}{2} \| c_j \| \sigma_r \| d \|^2,
\]

(19)

or an equivalent augmented linear system

\[
\begin{pmatrix} I & \nabla c_j^T \\ \nabla c_j & -\| c_j \|\sigma_r I \end{pmatrix} \begin{pmatrix} \xi \\ -d \end{pmatrix} = \begin{pmatrix} c_j \\ 0 \end{pmatrix}
\]

(20)

where \( \sigma_r \in [1, 2] \). An Armijo line search is used to compute \( \alpha_j = 2^{-l_j} \), where \( l_j \) is the smallest nonnegative integer satisfying

\[
\| c(x_j + 2^{-l_j} d_j) \|^2 \leq (1 - \rho) \| c_j \|^2 + \rho \| c_j + 2^{-l_j} \nabla c_j^T d_j \|^2.
\]

(21)

Note that subproblem (19) is a convex problem, hence can be easily solved. Moreover, nice convergence properties of Levenberg-Marquardt methods for degenerate problems have been shown by researchers, see, for instance, [59, 9, 60, 13] and etc.

Algorithm 2 Restoration Algorithm

Initialization. Given the iterate \( x_k \) at which the restoration phase is called and the maximal infeasibility \( h_k^{\text{max}} \). Set \( j := 0 \). Let \( x_j = x_k, c_j = c_k \) and \( \nabla c_j = \nabla c_k \).

repeat

Solve (19) or (20) to get search direction \( d_j \).

Determine step size \( \alpha_j \) that satisfies (21) using Armijo linesearch.

Let \( x_{j+1} = x_j + \alpha_j d_j, h_{j+1} = \| c_{j+1} \| \).

Set \( j := j + 1 \).

until \( h_j \leq h_k^{\text{max}} \)
2.4. The whole algorithm. Now we are ready to give the detailed description of the whole algorithm, see Algorithm 3.

Algorithm 3  Main Algorithm

Choose the following parameters

\[ 0 < \nu < \bar{\nu}, \gamma_v > 0, \gamma_1 > 0, \sigma_1 > 0, \gamma_2 \in (0, 1), \sigma_{\text{min}} \in (0, 1), \]
\[ \gamma_{\text{min}} > 0, \sigma_v > 2, \rho \in (0, 1), \alpha \in (0, 1), \gamma_\alpha \in (0, \frac{1}{2}), \]
\[ \sigma_\alpha > 1, \sigma_\beta > 0, \gamma_p > 0, \kappa_h, \bar{\kappa}_h \in (0, 1), \sigma_r \in [1, 2], \kappa_r \in (0, 1). \]

Initialize the primal dual entry as \((x_0, y_0)\). Let \(h_0^{\text{max}} = \max \{h_0, 10\}\).

Set \(k := 0\).

repeat

Get step \(d_k\) and style of line search \(LS_{\text{type}}\) using Algorithm 1. Use (16) to compute \(\alpha_k^{\text{min}}\).

if \(LS_{\text{type}} = 1\) then

Let \(\alpha_k\) be the largest scalar in \(\{2^{-j} : j = 1, 2, \ldots\}\)

which lies in \((\alpha_k^{\text{min}}, 1]\) and which satisfies (12) and (13).

else

Let \(\alpha_k\) be the largest scalar in \(\{2^{-j} : j = 1, 2, \ldots\}\)

which lies in \((\alpha_k^{\text{min}}, 1]\) and which satisfies (14).

end if

If the line search procedure above fails, quit Algorithm 3. Update \(h_k^{\text{max}}\) via (17). Enter the restoration phase (Algorithm 2).

until \(\|d_k\| + h_k \leq \epsilon\)

3. Global convergence. At the beginning of the discussion on global convergence, we consider the possible output of the restoration phase. By the convergence results of [59, 60, 13], the restoration phase, which applies Levenberg-Marquardt methods to (18), may terminate with one of the following two situations:

(a) finding a point \(x_r^k\) that approximately solving (18) in the sense of satisfying \(h(x_r^k) \leq h_k^{\text{max}}\);

(b) generating an (approximate) infeasible stationary point \(\bar{x}\) which satisfies \(\nabla c(x)c(x) = 0\).

In the second case, the restoration algorithm fails to get close enough to the feasible region. Hence, the whole algorithm cannot go any further. As Fletcher and Leyffer [18] pointed out, this situation always indicates that the problem is locally infeasible. Therefore, if case (b) happens, then the algorithm stops and reports local infeasibility.

From now on, we assume that the restoration phase always stops with case (a). To continue the argument on global convergence, we need the following assumptions.

Assumption 3.1. \(\text{Let } \{x_k\} \text{ be the sequence of the iterates generated by algorithm 3.}\)
(A1) There is a bounded convex set $\Omega$ containing $\{x_k\}$.
(A2) The gradients $\nabla f_k$ and $\nabla c_k$ are Lipschitz continuous on $\Omega$.
(A3) The symmetric matrix $H_k$ is uniformly bounded.
(A4) At any feasible accumulation point $\bar{x}$ of $\{x_k\}$, there is a neighborhood $\tilde{N}$ of it such that
(A4.a) the linear system
$$c_k + \nabla c_k^T p = 0$$
is consistent for any $x_k \in \tilde{N}$;
(A4.b) (Relaxed Constant Rank Constraint Qualification (RCRCQ)) the gradient $\nabla c_k$ has the same rank for all $x_k \in \tilde{N}$; and
(A4.c) (Second Order Sufficient Condition (SOSC)) there is a constant $\eta_1 > 0$ such that for any $x_k \in \tilde{N}$ the matrix $H_k$ is positive definite on the null space of $\nabla c_k^T$ in the sense of
$$p^T H_k p \geq \eta_1 \|p\|^2, \quad \forall p \in \{w \mid \nabla c_k^T w = 0\}$$
with $\eta_1 > 0$ fixed.

Of all the assumptions, (A1)-(A3) are standard for nonlinear constrained optimization. Assumptions (A4.a) and (A4.b) are weaker than the linear independence constraint qualification (LICQ), which requires that $\nabla c_k$ has full rank. Assumption (A4.c) is a usual assumption for line search algorithms. Hence, these assumptions are reasonable and weaker than standard assumptions.

3.1. Preliminary results. For the sake of simplifying notations, we use $M$ to denote a uniformly upper bound of $f_k$, $c_k$, $\nabla f_k$, $\nabla c_k$ and $H_k$. Without loss of generality, we assume $M > 1$.

Firstly, we consider the inertia of $K_k(\delta_k, v_k)$ under SOSC. We begin by recalling a result which is a variant of [29, Lemma 3].

Lemma 3.2. Suppose that Assumption 3.1 holds. Then there exist a positive constant, without loss of generality, $\eta_1$, a neighborhood $\tilde{N}$ of $\bar{x}$ and a positive $\hat{v}$ such that
$$p^T (H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T) p \geq \eta_1 \|p\|^2, \quad \forall p \in \mathbb{R}^n$$
holds for all $x_k \in \tilde{N}$ and $v_k \in (0, \hat{v})$.

Next, we show the relations between the SOSC and the inertia of $K_k(\delta_k, v_k)$.

Theorem 3.3. For the symmetric matrix $H_k$, the following two statements are equivalent:
(a) Assumption (A4.c) holds;
(b) there exists a positive constant which, with a slight abuse of notation, is also denoted by $\hat{v}$, such that $\ln(K_k(0, v_k)) = (n, m, 0)$ holds for all $x_k \in \tilde{N}$ and $v_k \in (0, \hat{v})$.

Proof. For $x_k \in \tilde{N}$, suppose that $\text{rank}(\nabla c_k) = r$. We write its singular value decomposition (SVD) as
$$\nabla c_k = \begin{pmatrix} \hat{U}_k & \hat{V}_k \end{pmatrix} \begin{pmatrix} \Sigma_k & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} U_k^T \\ V_k^T \end{pmatrix},$$

(23)
Then we have

\[ Q_k = \begin{pmatrix} \hat{U}_k & \hat{V}_k & 0 & 0 \\ 0 & 0 & U_k & V_k \end{pmatrix}. \]

Then we have

\[ Q_k^T K_k(0, v_k) Q_k = \begin{pmatrix} \hat{U}_k^T H_k \hat{U}_k & \hat{U}_k^T H_k \hat{V}_k & \hat{U}_k^T \nabla c_k U_k & \hat{U}_k^T \nabla c_k V_k \\ \hat{V}_k^T H_k \hat{U}_k & \hat{V}_k^T H_k \hat{V}_k & \hat{V}_k^T \nabla c_k U_k & \hat{V}_k^T \nabla c_k V_k \\ U_k^T \nabla c_k^T U_k & U_k^T \nabla c_k^T V_k & -v_k I_r & 0 \\ V_k^T \nabla c_k^T U_k & V_k^T \nabla c_k^T V_k & 0 & -v_k I_{m-r} \end{pmatrix}. \]

By (23), we have that

\[ \hat{U}_k^T \nabla c_k U_k = \Sigma_k, \]
\[ \hat{U}_k^T \nabla c_k V_k = 0, \]
\[ \hat{V}_k^T \nabla c_k U_k = 0, \]
\[ \hat{V}_k^T \nabla c_k V_k = 0. \]

By substituting these equalities into (24), we obtain

\[ Q_k^T K_k(0, v_k) Q_k = \begin{pmatrix} \hat{U}_k^T H_k \hat{U}_k & \hat{U}_k^T H_k \hat{V}_k & \Sigma_k & 0 \\ \hat{V}_k^T H_k \hat{U}_k & \hat{V}_k^T H_k \hat{V}_k & 0 & -v_k I_r \\ \Sigma_k & 0 & -v_k I_r & 0 \\ 0 & 0 & 0 & -v_k I_{m-r} \end{pmatrix}. \]

Let

\[ \hat{K}_k = \begin{pmatrix} \hat{U}_k^T H_k \hat{U}_k & \hat{U}_k^T H_k \hat{V}_k & \Sigma_k \\ \hat{V}_k^T H_k \hat{U}_k & \hat{V}_k^T H_k \hat{V}_k & 0 \\ \Sigma_k & 0 & -v_k I_r \end{pmatrix} \]

and

\[ R_k = \begin{pmatrix} I_r & 0 & \frac{1}{v_k} \Sigma_k \\ 0 & I_{n-r} & 0 \\ 0 & 0 & I_r \end{pmatrix}. \]

Then

\[ R_k \hat{K}_k R_k^T = \begin{pmatrix} \hat{U}_k^T H_k \hat{U}_k + \frac{1}{v_k} \Sigma_k^2 & \hat{U}_k^T H_k \hat{V}_k & 0 \\ \hat{V}_k^T H_k \hat{U}_k & \hat{V}_k^T H_k \hat{V}_k & 0 \\ 0 & 0 & -v_k I_r \end{pmatrix}. \]

Pre-multiplying and post-multiplying

\[ H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T \]
by \((\hat{U}_k \, \hat{V}_k)^T\) and \((\hat{U}_k \, \hat{V}_k)\) respectively, we have
\[
\begin{pmatrix}
\hat{U}_k^T \\
\hat{V}_k^T
\end{pmatrix}
\begin{pmatrix}
H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T
\end{pmatrix}
\begin{pmatrix}
\hat{U}_k \\
\hat{V}_k
\end{pmatrix}
= \begin{pmatrix}
\hat{U}_k^T \\
\hat{V}_k^T
\end{pmatrix} H_k \begin{pmatrix}
\hat{U}_k \\
\hat{V}_k
\end{pmatrix} + \frac{1}{v_k} \begin{pmatrix}
\hat{U}_k^T \\
\hat{V}_k^T
\end{pmatrix} \nabla c_k \nabla c_k^T \begin{pmatrix}
\hat{U}_k \\
\hat{V}_k
\end{pmatrix}
= \begin{pmatrix}
\hat{U}_k^T \\
\hat{V}_k^T
\end{pmatrix} H_k \begin{pmatrix}
\hat{U}_k \\
\hat{V}_k
\end{pmatrix} + \frac{1}{v_k} \begin{pmatrix}
\hat{U}_k^T \\
\hat{V}_k^T
\end{pmatrix} \nabla c_k \begin{pmatrix}
U_k \\
V_k
\end{pmatrix} \begin{pmatrix}
\hat{U}_k^T \\
\hat{V}_k^T
\end{pmatrix} \nabla c_k^T \begin{pmatrix}
\hat{U}_k \\
\hat{V}_k
\end{pmatrix}
= \begin{pmatrix}
\hat{U}_k^T H_k \hat{U}_k + \frac{1}{v_k} \Sigma^2_k \\
V_k^T H_k \hat{V}_k
\end{pmatrix}
\begin{pmatrix}
\hat{U}_k^T H_k \hat{V}_k \\
V_k^T H_k \hat{V}_k
\end{pmatrix}
\]

Since \((\hat{U}_k \, \hat{V}_k)\) is orthonormal and by Lemma 3.2, the symmetric matrix
\[
\tilde{K}_k = \begin{pmatrix}
\hat{U}_k^T H_k \hat{U}_k + \frac{1}{v_k} \Sigma^2_k & \hat{U}_k^T H_k \hat{V}_k \\
V_k^T H_k \hat{U}_k & V_k^T H_k \hat{V}_k
\end{pmatrix}
\]
is also positive definite. Then it follows from the decomposition (27) that, \(\text{In}(\tilde{K}_k) = (n, r, 0)\), and then from (26) that \(\text{In}(K_k(0, v_k)) = (n, m, 0)\). Hence, we prove that (a) \(\Rightarrow\) (b).

Next, we are going to prove the reverse implication. Suppose that \(\text{In}(K_k(0, v_k)) = (n, m, 0)\). Using the decomposition (26) and (27), we conclude that the matrix \(K_k\) is positive definite. Choose \(\hat{v} > 0\) small enough such that the (1,1) block of the above matrix is positive definite for all \(v_k \in (0, \hat{v})\). Let
\[
J_k = \begin{pmatrix}
I_r & - (\hat{U}_k^T H_k \hat{U}_k + \frac{1}{v_k} \Sigma^2_k)^{-1} \hat{U}_k^T H_k \hat{V}_k \\
0 & I_{n-r}
\end{pmatrix}
\]
Then
\[
J_k^T K_k J_k = J_k^T \begin{pmatrix}
\hat{U}_k^T H_k \hat{U}_k + \frac{1}{v_k} \Sigma^2_k & \hat{U}_k^T H_k \hat{V}_k \\
V_k^T H_k \hat{U}_k & V_k^T H_k \hat{V}_k
\end{pmatrix} J_k
= \begin{pmatrix}
\hat{U}_k^T H_k \hat{U}_k + \frac{1}{v_k} \Sigma^2_k & 0 \\
0 & V_k^T H_k \hat{V}_k - v_k \hat{V}_k^T H_k \hat{U}_k (v_k I_r + \Sigma^2_k)^{-1} \hat{U}_k^T H_k \hat{V}_k
\end{pmatrix}
\]
Note that the positive definiteness of \(K_k\) implies that
\[
\hat{V}_k^T H_k \hat{V}_k - v_k \hat{V}_k^T H_k \hat{U}_k (v_k I_r + \Sigma^2_k)^{-1} \hat{U}_k^T H_k \hat{V}_k
\]
is positive definite for all \(v_k \in (0, \hat{v})\). Therefore, the matrix \(\hat{V}_k^T H_k \hat{V}_k\) must be positive definite, which is equivalent to (a). The proof is done.

This theorem indicates that when \(x_k\) is close enough to a feasible point, the primal regularization parameter \(\delta_k\) equals 0. When the point is far from the feasible region, a proper \(\delta_k\) can be found because of the following lemma

**Lemma 3.4.** [24, Lem. 2.2(ii)] For a given \(\delta_k \geq 0\), the matrix \(H_k + \delta_k I + \frac{1}{v_k} \nabla c_k \nabla c_k^T\) is positive definite if and only if \(\text{In}(K_k(\delta_k, v_k)) = (n, m, 0)\).
Without loss of generality, we assume $\delta_k = 0$ from now on. For convenience, we abbreviate $K_k(0, v_k)$ as $K_k$. Next, we show the boundedness of $y_k^+$ near a feasible point.

Lemma 3.5. Suppose that Assumptions 3.1 hold. Let $\bar{x}$ is a feasible point. Then there is a neighborhood $\tilde{N}$ of $\bar{x}$ such that, by reducing $\hat{\nu}$ if necessary, the set \{ $y_k^+ | x_k \in \tilde{N}$ \} is bounded for all $v_k \in (0, \hat{\nu})$.

Proof. Let $s_k$ be the minimal $\ell_2$ norm solution of (22), i.e., $s_k$ solves

$$\min \| p \|$$
$$\text{s.t.} \quad c_k + \nabla c_k^T p = 0.$$ 

Then, by (A4), $s_k$ is well defined and by Assumptions 3.1 and [47, Theorem 1.2.7], $s_k$ is bounded near a feasible point. Let $t_k = d_k - s_k$. By substituting it into (6), we have

$$\begin{pmatrix} H_k & \nabla c_k \\ \nabla c_k^T & -v_k I \end{pmatrix} \begin{pmatrix} t_k \\ y_k^+ \end{pmatrix} = -\begin{pmatrix} \nabla f_k + H_k s_k \\ 0 \end{pmatrix}. \quad (28)$$

To analyze the step $(t_k, y_k^+)$, we decompose it according to the SVD (23) as

$$t_k = \hat{U}_k p_{\hat{U}_k} + \hat{V}_k p_{\hat{V}_k},$$
$$y_k^+ = U_k z_{U_k} + V_k z_{V_k}. \quad (29)$$

By substituting (29) into (28), pre-multiplying the blocks of this system by $\hat{U}_k^T$, $\hat{V}_k^T$, $U_k^T$ and $V_k^T$, and using (25), we botain that

$$\begin{pmatrix} \hat{U}_k^T H_k \hat{U}_k & \hat{U}_k^T H_k \hat{V}_k & \Sigma_k & 0 \\ \hat{V}_k^T H_k \hat{U}_k & \hat{V}_k^T H_k \hat{V}_k & 0 & 0 \\ \Sigma_k & 0 & -v_k I_r & 0 \\ 0 & 0 & 0 & -v_k I_{m-r} \end{pmatrix} \begin{pmatrix} p_{\hat{U}_k} \\ p_{\hat{V}_k} \\ z_{U_k} \\ z_{V_k} \end{pmatrix} = \begin{pmatrix} r_{\hat{U}_k} \\ r_{\hat{V}_k} \\ 0 \end{pmatrix}. \quad (30)$$

where $r_{\hat{U}_k} = -\hat{U}_k^T (\nabla f_k + H_k s_k)$, $r_{\hat{V}_k} = -\hat{V}_k^T (\nabla f_k + H_k s_k)$. Choose a proper neighborhood $\tilde{N}$ of $\bar{x}$ such that the singular value $\sigma_k^{(r)} > \hat{\sigma} > 0$ and that

$$w^T \hat{V}_k^T H_k \hat{V}_k w \geq \eta_1 \| w \|^2, \forall \ w \in R^{n-r}$$

with some fixed $\hat{\sigma} > 0$ for all $x_k \in \tilde{N}$. By the fourth row of (30), we have $z_{V_k} = 0$ and

$$\begin{pmatrix} \hat{U}_k^T H_k \hat{U}_k & \hat{U}_k^T H_k \hat{V}_k & \Sigma_k \\ \hat{V}_k^T H_k \hat{U}_k & \hat{V}_k^T H_k \hat{V}_k & 0 \\ \Sigma_k & 0 & -v_k I_r \end{pmatrix} \begin{pmatrix} p_{\hat{U}_k} \\ p_{\hat{V}_k} \\ z_{U_k} \end{pmatrix} = \begin{pmatrix} r_{\hat{U}_k} \\ r_{\hat{V}_k} \\ 0 \end{pmatrix}. \quad (31)$$

By similar arguments as [56, Section 3.1] and possibly shrinking $\tilde{N}$, we have that, reducing $\hat{\nu}$ if necessary, the solution of (31) is bounded for all $x_k \in \tilde{N}$ and $v_k \in (0, \hat{\nu})$. Particularly, there is a positive $M_y$ such that

$$\| y_k^+ \| = \| U_k z_{U_k} + V_k z_{V_k} \| \leq \| z_{U_k} \| \leq M_y$$

in this case.

By compactness, we have

Corollary 3.6. Suppose that Assumptions 3.1 hold. Then the sequence \{ $y_k^+$ \} is bounded.
Lemma 3.7. Let \( \hat{x} \) and \( \hat{\mathcal{N}} \) be a feasible accumulation point of \( \{x_k\} \) and its neighborhood that satisfy (A4). Let \( \hat{v} \) be the constant given in Theorem 3.3. Then for all \( x_k \in \hat{\mathcal{N}} \) and \( v_k \in [0, \hat{v}] \), the following inequalities hold

\[
v_k \eta_1 \leq v_k \lambda_{\min}^{H_k} + (\sigma_{\max} c_k)^2 \leq v_k \lambda_{\max}^{H_k} + (\sigma_{\max} c_k)^2 \quad (33)
\]

Proof. Let \( \xi \neq 0 \) be an eigenvector of \( H_k \) associated with \( \lambda_{\min}^{H_k} \). It follows from Lemma 3.2 that

\[
v_k \eta_1 \leq v_k \xi^T H_k \xi + \xi^T \nabla c_k \nabla c_k^T \xi \quad \xi^T \xi
\]

\[
= v_k \lambda_{\min}^{H_k} \|\xi\|^2 + \xi^T \nabla c_k \nabla c_k^T \xi \|\xi\|^2
\]

\[
\leq v_k \lambda_{\min}^{H_k} \|\xi\|^2 + (\sigma_{\max} c_k)^2 \|\xi\|^2
\]

\[
= v_k \lambda_{\min}^{H_k} + (\sigma_{\max} c_k)^2.
\]

The second inequality of (33) follows immediately from the first inequality and \( \lambda_{\min}^{H_k} \leq \lambda_{\max}^{H_k} \).

Lemma 3.7 is useful for the following result, which is inspired by Rusten and Winther [44, Lem. 2.1], Silvester and Wathen [46, Lem. 2.2], Gondzio [26, Lem. 3.1] and Friedlander and Orban [20, Thm. 5.1].

Theorem 3.8. Suppose that \( \{x_k\} \), \( \hat{x} \) and \( \hat{\mathcal{N}} \) satisfies Assumption 3.1 and that \( \hat{v} \) be the constant given in Theorem 3.3. For any \( x_k \in \hat{\mathcal{N}} \) and \( 0 < v_k \leq \hat{v} \), the matrix \( K_k \) possesses precisely \( n \) positive eigenvalues and \( m \) negative eigenvalues. Let them be denoted and ordered as

\[
\lambda_{-m} \leq \lambda_{-m+1} \leq \cdots \leq \lambda_{-1} < 0 < \lambda_1 \leq \cdots \lambda_{n-1} \leq \lambda_n.
\]

Then

\[
\lambda_{-m} \geq \frac{1}{2} \left( \lambda_{\min}^{H_k} - v_k - \sqrt{(\lambda_{\min}^{H_k} - v_k)^2 + 4(v_k \lambda_{\min}^{H_k} + (\sigma_{\max} c_k)^2)} \right). \quad (34)
\]

\[
\lambda_n \leq \frac{1}{2} \left( \lambda_{\max}^{H_k} - v_k + \sqrt{(\lambda_{\max}^{H_k} - v_k)^2 + 4(v_k \lambda_{\max}^{H_k} + (\sigma_{\max} c_k)^2)} \right). \quad (35)
\]

and

\[
\lambda_1 \geq \min \left\{ \eta_1, \frac{1}{2} \left( \eta_1 - \hat{v} - \frac{(\sigma_{\max} c_k)^2}{\hat{v}} + \sqrt{(\hat{v} - \frac{(\sigma_{\max} c_k)^2}{\hat{v}} + \eta_1)^2 + 4(\sigma_{\max} c_k)^2} \right) \right\}. \quad (36)
\]

Moreover, the smallest positive eigenvalue \( \lambda_1 \) is bounded away from 0 for all \( x_k \in \hat{\mathcal{N}} \) and \( 0 < v_k \leq \hat{v} \), i.e., there is a fixed \( \hat{\lambda} \) such that

\[
\lambda_1 \geq \hat{\lambda} > 0, \ \forall x_k \in \hat{\mathcal{N}}, \ 0 < v_k \leq \hat{v}. \quad (38)
\]

Proof. If \( (\xi, \eta) \neq 0 \) is an eigenvector of \( K_k \) associated with the eigenvalue \( \lambda \), then

\[
H_k \xi + \nabla c_k \eta = \lambda \xi, \quad (39)
\]

\[
\nabla c_k^T \xi - v_k \eta = \lambda \eta. \quad (40)
\]

Note that \( \lambda = -v_k < 0 \) is an eigenvalue of \( K_k \) if and only if \( \nabla c_k \) does not have full column rank, and in this case, its associated eigenspace is \( \{0\} \times \text{Null}(\nabla c) \).
Now suppose $\lambda \neq -v_k$. From (40), we have $\eta = (\lambda + v_k)^{-1}\nabla c_k^T \xi$. Necessarily, $\xi \neq 0$. Substituting into (39) and taking the inner product with $\xi$ yields
\[
\lambda \|\xi\|^2 = \xi^T H_k \xi + (\lambda + v_k)^{-1}\xi^T \nabla c_k \nabla c_k^T \xi.
\] (41)

If $-v_k < \lambda < 0$, then by Lemma 3.2, the right-hand side of (41) satisfies
\[
\xi^T H_k \xi + (\lambda + v_k)^{-1}\xi^T \nabla c_k \nabla c_k^T \xi \geq \xi^T (H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T) \xi \geq \eta \|\xi\|^2 > 0,
\]
which contradicts $\lambda < 0$. Hence, either $\lambda \leq -v_k$ or $\lambda \geq 0$. In particular, $\lambda \leq -v_k$ must be true of $\lambda_{-1}$, which implies (36).

If $\lambda < -v_k$, we deduce from (41) that
\[
\lambda \|\xi\|^2 \geq \lambda_{\min} \|\xi\|^2 + (\lambda + v_k)^{-1}(\sigma_{\max}^2) \|\xi\|^2.
\]

Upon simplifying and substituting $z$ for $\lambda$, we see that the quadratic
\[
z^2 + z(v_k - \frac{H_k}{\lambda_{\min}}) - (v_k \lambda_{\min} + (\sigma_{\max}^2))
\]
in $z$ takes a non-positive value when evaluated at an eigenvalue $\lambda < -v_k$ of $K_k$. In particular, this must be true of $\lambda_{-m}$, which implies (34).

If $\lambda > 0$, then we deduce from (41) that
\[
\lambda \|\xi\|^2 \leq \lambda_{\max} \|\xi\|^2 + (\lambda + v_k)^{-1}(\sigma_{\max}^2) \|\xi\|^2.
\]

This implies that the quadratic
\[
z^2 + z(v_k - \frac{H_k}{\lambda_{\max}}) - (v_k \lambda_{\max} + (\sigma_{\max}^2))
\]
takes non-positive value when evaluated at an eigenvalue $\lambda > 0$ of $K_k$, which implies (35).

Now, we consider (37). For a positive eigenvalue $\lambda$, we have, by (41) and Lemma 3.2, that
\[
\lambda \|\xi\|^2 = \xi^T H_k \xi + (\lambda + v_k)^{-1}\xi^T \nabla c_k \nabla c_k^T \xi
\]
\[
= \xi^T H_k \xi + \hat{\nu}^{-1}\xi^T \nabla c_k \nabla c_k^T \xi + ((\lambda + v_k)^{-1} - \hat{\nu}^{-1})\xi^T \nabla c_k \nabla c_k^T \xi.
\] (42)

If $(\lambda + v_k)^{-1} - \hat{\nu}^{-1} \geq 0$, then it follows that
\[
\lambda \|\xi\|^2 \geq \xi^T H_k \xi + \hat{\nu}^{-1}\xi^T \nabla c_k \nabla c_k^T \xi \geq \eta \|\xi\|^2,
\]
which implies that $\lambda \geq \eta_1$. Otherwise, it follows (42) that
\[
\lambda \|\xi\|^2 \geq \eta_1 \|\xi\|^2 + ((\lambda + v_k)^{-1} - \hat{\nu}^{-1})(\sigma_{\max}^2) \|\xi\|^2.
\]

In turn, this implies that the quadratic
\[
z^2 + z(v_k + (\sigma_{\max}^2)\hat{\nu} - \eta_1) - (\eta_1 - (\sigma_{\max}^2)\hat{\nu}) v_k - (\sigma_{\max}^2)
\] (43)
in $z$ takes a nonnegative value when evaluated at an eigenvalue $\lambda > 0$ of $K_k$, which implies
\[
\lambda \geq \frac{1}{2} \left( \eta_1 - v_k - (\sigma_{\max}^2)\hat{\nu} + \sqrt{(v_k - (\sigma_{\max}^2)\hat{\nu} + \eta_1)^2 + 4(\sigma_{\max}^2)} \right).
\] (44)

Note that the right-hand side of (44) is monotonically decreasing on $v_k$. Then, a positive eigenvalue of $K_k$ satisfies the following lower bound
\[
\lambda \geq \frac{1}{2} \left( \eta_1 - \hat{\nu} - (\sigma_{\max}^2)\hat{\nu} + \sqrt{((\hat{\nu} - (\sigma_{\max}^2)\hat{\nu} + \eta_1)^2 + 4(\sigma_{\max}^2)} \right).
\]
Therefore, we have

\[
\lambda \geq \min \left\{ \eta_1, \frac{1}{2} \left( \eta_1 - \hat{v} - (\sigma_{\text{max}})^2 / \hat{v} + \sqrt{\left( \hat{v} - (\sigma_{\text{max}})^2 / \hat{v} + \eta_1 \right)^2 + 4(\sigma_{\text{max}})^2} \right) \right\}.
\]

In particular this must be true of \( \lambda_1 \), which implies (37).

This lower bound is positive and bounded away from 0 because in (43), the last term

\[-(\eta_1 - (\sigma_{\text{max}})^2 / \hat{v}) v_k - (\sigma_{\text{max}})^2 = -\eta_1 - (1 - v_k / \hat{v})(\sigma_{\text{max}})^2 \leq -\eta_1 < 0\]

and the coefficient \( v_k + (\sigma_{\text{max}})^2 / \hat{v} - \eta_1 \) is bounded for all \( x_k \in \tilde{N} \) and \( v_k \in (0, \hat{v}] \).

Thus, there is a positive \( \hat{\lambda} \) such that (38) holds.

Two immediate consequences of this theorem are as following, which provide upper bounds of \( \{K_{k}^{-1}\} \) and \( \|d_k\| \) with respect to \( v_k \).

**Corollary 3.9.** Let \( \tilde{x} \) and \( \tilde{N} \) be a feasible accumulation point and its neighborhood that satisfy (A4) and (A5). For all \( x_k \in \tilde{N} \) and \( v_k \in (0, \hat{v}] \),

\[\|K_k^{-1}\| \leq \frac{1}{\min\{\hat{\lambda}, v_k\}}.\]

**Corollary 3.10.** Let \( \tilde{x} \) and \( \tilde{N} \) be a feasible accumulation point and its neighborhood that satisfy (A4) and (A5). For all \( x_k \in N \) and \( v_k \in (0, \hat{v}] \),

\[\|d_k\| \leq \frac{\|\nabla f_k\| + \|c_k\|}{\min\{\hat{\lambda}, v_k\}}.\]

If \( v_k \) is sufficiently small, then

\[\|d_k\| \leq \frac{1}{v_k} (\|\nabla f_k\| + \|c_k\|).\]

### 3.2. Well-definedness.

By the mechanism of the algorithms, both Algorithm 1 and Algorithm 3 are well-defined because both algorithms turn to the restoration phase when the inner loops encounter too small parameters (\( v_k \) in Algorithm 1 and \( \alpha_k \) in Algorithm 3). However, restoration phase near the solution region is undesirable because it leads to feasible points instead of stationary ones. In this subsection, we will show that, under Assumption 3.1, this undesirable situation does not happen when the iterate is close enough to the feasible region.

**Lemma 3.11.** Let \( \tilde{x} \) and \( \tilde{N} \) be a feasible accumulation point and its neighborhood that satisfy (A4). Then, by choosing \( v_k \) small enough, either the corresponding step \( y_k^+ \) satisfies (9), or it satisfies (10).

**Proof.** It follows directly from Lemma 3.5 and Corollary 3.6 that (9) holds if

\[v_k \leq \frac{\gamma_2 h_{k}^{\text{max}}}{M_y},\]  

and (10) holds if

\[v_k \leq \frac{\gamma_2 h_k}{M_y},\]

where \( M_y \) is the constant given in (32).

Thus, Algorithm 1 will not call restoration phase near a feasible point.
Lemma 3.12. Suppose that Assumptions 3.1 hold and that $\bar{x}$ is a feasible accumulation point of $\{x_k\}$. Then there is a neighborhood $\mathcal{N}$ of $\bar{x}$, such that Algorithm 1 terminates with a regularization parameter $v_k \geq v_k^{\min}$ for all $x_k \in \mathcal{N}$, i.e., the restoration algorithm is not called.

Proof. By (11), we have $v_k^{\min} = o(h_k)$. Thus, this Lemma is trivially true because of (45) and (46).

Next, we consider the finite termination of the Armijo line search in Algorithm 3.

Lemma 3.13. Suppose that the Algorithm 3 does not terminate at $x_k$ and that $v_k$ is chosen such that (8) and (9) are satisfied. Then there exists a positive scalar $\alpha^f_k$ such that for any $\alpha \in (0, \alpha^f_k]$, both (12) and (13) are satisfied.

Proof. It is trivial that $-\nabla f_k^T d_k > 0$ in the case $h_k > 0$ because of (8). If $h_k = 0$, then
$$\left( \begin{array}{c} H_k \\ \nabla c_k^T \\ -v_k I \end{array} \right) \left( \begin{array}{c} d_k \\ y_k^+ \end{array} \right) = -\left( \begin{array}{c} \nabla f_k \\ 0 \end{array} \right).$$

By eliminating $y_k^+$, we get
$$H_k d_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T d_k = -\nabla f_k.$$

Taking inner product with $d_k$ and by Lemma 3.2, we obtain
$$-\nabla f_k^T d_k = d_k^T \left( H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T \right) d_k \geq \eta_1 \|d_k\|^2 > 0.$$

Using standard Taylor’s Theorem and Lipschitz continuity, we have, with a Lipschitz constant $L_{df}$,
$$|f_k - f(x_k + \alpha d_k) - (-\alpha \nabla f_k^T d_k)| \leq \alpha^2 L_{df} \|d_k\|^2.$$

Using this inequality, we deduce
$$\frac{|f_k - f(x_k + \alpha d_k) - (-\alpha \nabla f_k^T d_k)|}{-\alpha \nabla f_k^T d_k} \leq \frac{L_{df} \|d_k\|^2}{-\nabla f_k^T d_k}.$$ 

Therefore, (12) holds for any step size $\alpha$ that satisfies
$$0 < \alpha \leq \frac{- (1 - \rho) \nabla f_k^T d_k}{L_{df} \|d_k\|^2}.$$

Next, using convexity of $\ell_2$ norm, after similar arguments, (13) is satisfies for any step size satisfying
$$0 < \alpha \leq \frac{(1 - \gamma_2) h_k^{\max}}{L_{dc} \|d_k\|^2}.$$

with $L_{dc}$ a Lipschitz constant for $\nabla c$.

To sum up, the statement is true with
$$\alpha^f_k := \min \left\{ \frac{- (1 - \rho) \nabla f_k^T d_k}{L_{df} \|d_k\|^2}, \frac{(1 - \gamma_2) h_k^{\max}}{L_{dc} \|d_k\|^2} \right\}. \quad (47)$$

In the other case, there is a similar result.
Lemma 3.14. Suppose that the Algorithm 3 does not terminate at \( x_k \) and \( v_k \) is chosen such that (10) holds. Let
\[
\alpha^h_k := \frac{(1 - \rho)(1 - \gamma_2)h_k}{2L_{dc}\|d_k\|^2}.
\]
Then for any \( \alpha \in (0, \alpha^h_k] \), the condition (14) is satisfied.

Now, Lemmas 3.13 and 3.14 indicate that, under Assumption 3.1 and by (16), the line search will terminate successfully and the restoration algorithm will not be called if the iterate is close enough to the feasible region.

Lemma 3.15. Suppose that \( \tilde{x} \) is a feasible accumulation point of \( \{x_k\} \). Then there is a neighborhood \( \tilde{N} \) of \( \tilde{x} \), such that Algorithm 3 does not call restoration algorithm at all \( x_k \in \tilde{N} \).

Proof. By (16), we have that \( \alpha^h_{\min} = \alpha^f_kO(h^\sigma_{\beta}) \) for an \( f \)-iteration and \( \alpha^h_{\min} = o(h_k) \) for an \( h \)-iteration. Then this statement is true because of Lemmas 3.13 and 3.14. \( \square \)

3.3. Global convergence theorem. Now, we are ready to present the global convergence results for the presented algorithm. These results parallel convergence discussions in [43], with main difference that the regularization parameter \( v_k \) is concerned here, which causes some complexity in the proofs. We shall use the following two index sets
\[
K_h = \{k \mid h_{k+1}^{\max} < h_k^{\max}\}, \quad \bar{K}_h = \{k \mid h_{k+1}^{\max} = \kappa_h h_k^{\max} \text{ or } h_{k+1}^{\max} = \kappa_r h_k^{\max}\},
\]

First, we recall a result concerning a direct consequence of the definition of \( h_{k+1}^{\max} \).

Lemma 3.16. [27] The sequence \( \{h_k^{\max}\} \) is non-increasing and the inequality
\[
0 \leq h_l \leq h_k^{\max}
\]
holds for all \( l \geq k \).

The main result we will establish is the following.

Theorem 3.17. Suppose that Algorithm 3 does not terminate finitely, and that Assumptions 3.1 hold.

(1) If \( |K_h| = \infty \), then \( \lim_{k \to +\infty} h_k = 0 \), \( \lim_{k \in \bar{K}_h} \|d_k\| = 0 \).

(2) If \( |K_h| < \infty \), then \( \lim_{k \to +\infty} h_k = 0 \), \( \lim_{k \to +\infty} \|d_k\| = 0 \).

We shall prove this theorem by proving a series of lemmas. Firstly, we show convergence to feasible region by proving two lemmas.

Lemma 3.18. Suppose that Algorithm 3 does not terminate finitely, that Assumptions 3.1 hold and that \( |\bar{K}_h| = \infty \). Then \( \lim_{k \to +\infty} h_k = 0 \).

Proof. We first consider the case where \( |\bar{K}_h| = \infty \). By the definition of \( \bar{K}_h \) and Lemma 3.16, we have
\[
0 \leq \lim_{k \to +\infty} h_k \leq \lim_{k \to +\infty} h_k^{\max} = 0,
\]
which implies \( \lim_{k \to +\infty} h_k = 0 \).

Next, we consider the other case where \( |\bar{K}_h| < \infty \). In this case, it follows from the definition of \( \bar{K}_h \) and (15) that
\[
h_{k+1}^{\max} = \tilde{\kappa}_h h_k + (1 - \tilde{\kappa}_h)h_{k+1}
\]
for sufficiently large $k$. Using this and Lemma 3.16, we get
\[ h^\text{max}_k - h^\text{max}_{k+1} \geq (1 - \bar{\kappa}_k)(h_k - h^\text{max}_{k+1}) \]
\[ \geq (1 - \bar{\kappa}_k)\rho \alpha_k h_k. \]

The last inequality combined with Lemma 3.14 yields
\[ h^\text{max}_k - h^\text{max}_{k+1} \geq \rho(1 - \bar{\kappa}_k)(1 - \rho)(1 - \gamma_2)h_k^2. \]

Then it follows from Corollary 3.10 and (38) that
\[ \|d_k\| \leq \min\{\lambda, v_k\}(\|\nabla f_k\| + \|c_k\|) \leq \frac{2M}{\min\{\lambda, v_k\}}, \]
where $M$ is a positive constant defined at the beginning of Subsection 3.1. By Lemma 3.11 and (45), we have
\[ v_k \geq \min\left\{\frac{1}{2}\gamma_2 h_k^\text{max}, \frac{1}{2M_y}\right\}. \]

By (48), (49), (50) and the fact that $h^\text{max}_k - h^\text{max}_{k+1}$ tends to 0 because of Lemma 3.16, we have $\lim_{k \to +\infty} h_k = 0$. \(\square\)

**Lemma 3.19.** Suppose that Algorithm 3 does not terminate finitely, that Assumptions 3.1 hold and that $|K_h| < \infty$. Then $\lim_{k \to +\infty} h_k = 0$.

**Proof.** In this case, for sufficiently large $k$, (8) holds and $h^\text{max}_k = h^\text{max}_{\tilde{k}}$ with some fixed index $\tilde{k}$. Hence, we have
\[ f_k - f_{k+1} \geq -\rho \alpha_k \nabla f_k^T d_k \geq \rho \gamma_1 \alpha_k h^\text{max}_{\tilde{k}}. \]

By Lemma 3.13, (47), the mechanism of line search and (8),
\[ \alpha_k \geq \frac{1}{2} \alpha_k \geq \min\left\{\frac{(1 - \rho)\gamma_1 h^\text{max}_{\tilde{k}}}{L_d}, \frac{(1 - \gamma_2)h^\text{max}_{\tilde{k}}}{L_d}\right\} \frac{1}{\|d_k\|^2}. \]

By Lemma 3.11 and (45) and increasing $\tilde{k}$ if necessary, we have
\[ v_k \geq \min\left\{\frac{1}{2}\gamma_2 h^\text{max}_{\tilde{k}}, \frac{1}{2M_y}\right\}, \forall \tilde{k} \geq \tilde{k}. \]

Note that $\{f_k\}$ is monotonically decreasing except for finite points. Then by (51)-(52) and (49), we have $\lim_{k \to +\infty} h_k = 0$. \(\square\)

**Lemma 3.20.** Suppose that Algorithm 3 does not terminate finitely, that Assumptions 3.1 hold and that $|K_h| = \infty$. Then $\lim_{k \in K_h \in K_h} \|d_k\| = 0$.

**Proof.** By Lemmas 3.12, 3.15, 3.18 and (50), the algorithm does not call restoration algorithm for sufficiently large $k$. Without loss of generality, we assume that (8) does not hold at all $k \in K_h$. Then by eliminating $y^*_k$ in system (6), we have, for all $k \in K_h$,
\[ \eta_1 \|d_k\|^2 \leq d_k^T \left( H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T \right) d_k \]
\[ = -\nabla f_k^T d_k - \frac{1}{v_k} c_k^T \nabla c_k^T d_k. \]

(53)
Then, from the second row of (6) and (10), it follows that
\[
\eta_1 \|d_k\|^2 \leq \gamma_1 h_k^{\alpha_1} + \frac{(1 + \gamma_2) h_k^2}{v_k}.
\] (54)
It is noteworthy that by the mechanism of Algorithm 1 and (46), there is \(v_k \geq \frac{1}{2} \min\{v_k^e, \gamma_2 h_k / M_y\}\). Note that \(v_k^e \in [\underline{v}, \overline{v}]\). Hence, \(v_k^{-1} h_k\) is bounded. Then taking limit on both sides of (54) for \(k \in \mathcal{K}_h\) and using Lemma 3.18, we have \(\lim_{k \in \mathcal{K}_h} \|d_k\| = 0\). This completes the proof.

**Lemma 3.21.** Suppose that Algorithm 3 does not terminate finitely, that Assumptions 3.1 hold and that \(|\mathcal{K}_h| < +\infty\). Then \(\lim_{k \to +\infty} \|d_k\| = 0\).

**Proof.** In this case, condition (8) holds for all \(k \geq \bar{k}\), where \(\bar{k}\) is the constant defined in the proof of Lemma 3.19. We first prove \(\lim_{k \to +\infty} -\nabla f_k^T d_k = 0\) by contradiction. Suppose that there is an index set \(\mathcal{K}\) such that
\[
\lim_{k \in \mathcal{K}} -\nabla f_k^T d_k = \chi^* > 0.
\] (55)
Note that \(f_k - f_{k+1} \geq 0\) for all \(k \geq \bar{k}\). Then
\[
f_0 - f_{k+1} = \sum_{j=0}^{k-1} (f_j - f_{j+1})
= \sum_{j=0}^{k-1} (f_j - f_{j+1}) + \sum_{j=k}^{k-1} (f_j - f_{j+1})
\geq \sum_{j=0}^{k-1} (f_j - f_{j+1}) + \sum_{k \leq j, k \in \mathcal{K}} (f_j - f_{j+1})
\geq \sum_{j=0}^{k-1} (f_j - f_{j+1}) + \sum_{k \leq j, k \in \mathcal{K}} (-\rho \alpha_j \nabla f_j^T d_j).
\]
For all \(k \geq \bar{k}\), we have \(h_k^{\max} = h_k^{\max}\) by the rule of updating \(h_k^{\max}\). By Corollary 3.9, the coefficient matrix \(\{K_k^{-1}\}\) is bounded and in turn, by Corollary 3.10, so is \(\{d_k\}\). Without loss of generality, we assume that \(\|d_k\| \leq M\). By Algorithm 1, Lemma 3.11 and (45), we have \(v_k \geq \frac{1}{2} \left\{ \underline{v}, \frac{\gamma_2 h_k^{\max}}{M_y} \right\}\) for all \(k\). By (47), (49) and the fact that \(h_k^{\max}\) keeps unchanged for sufficiently large \(k\), the step size \(\alpha_j\) is bounded away from 0 for \(j \in \mathcal{K}\). Then by (55), the last term of above inequalities tends to +\(\infty\). However, under Assumptions (A1) and (A2), the term \(\sum_{k < j, k \in \mathcal{K}} (-\rho \alpha_j \nabla f_j^T d_j)\) is bounded, which yields a contradiction. Therefore, \(\lim_{k \to +\infty} -\nabla f_k^T d_k = 0\).

Thus, by (53), (32) and Lemma 3.19, we have
\[
\eta_1 \|d_k\|^2 \leq -\nabla f_k^T d_k + \frac{1}{v_k} c_k^T (c_k - v_k y_k^+)
\leq -\nabla f_k^T d_k + \frac{1}{2} \left\{ \underline{v}, \frac{\gamma_2 h_k^{\max}}{M_y} \right\} c_k^T y_k^+
\leq -\nabla f_k^T d_k + \frac{1}{2} \left\{ \underline{v}, \frac{\gamma_2 h_k^{\max}}{M_y} \right\} + M_y h_k,
\]
which implies \(\lim_{k \to +\infty} \|d_k\| = 0\). \(\square\)
To summarize the previous results, we have proved Theorem 3.17. Note that Theorem 3.17 implies convergence to KKT points.

**Lemma 3.22.** Suppose that Assumptions 3.1 hold, and that $\bar{K}$ is an index set which satisfies $\lim_{k \in \bar{K}} h_k = 0$ and $\lim_{k \in \bar{K}} d_k = 0$. Suppose also, without loss of generality that $\lim_{k \in \bar{K}} x_k = \tilde{x}$. Then $\tilde{x}$ is a KKT point.

**Proof.** By continuity, we have that $\tilde{x}$ is a feasible point. By eliminating $y_k^+$ in system (6), there is

$$
(H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T) d_k + \nabla f_k + \frac{1}{v_k} \nabla c_k c_k = 0,
$$

If $\{v_k\}$ is bounded away from 0, then by taking limit on both sides of this equality, we have that

$$
\nabla f(\tilde{x}) = \lim_{k \in \bar{K}} \nabla f_k = 0,
$$

which indicates that $\tilde{x}$ is a KKT point and 0 is the corresponding multiplier.

Otherwise, assume, without loss of generality that $\lim_{k \in \bar{K}} v_k = 0$. Then, by the mechanism of updating $v_k$, the index set $\mathcal{K}_h$ is infinite. By (10) and the second row of (6), we have

$$
\lim_{k \in \bar{K}} \|\nabla c_k^T d_k\| = 0.
$$

Since, by (50) and (52), the sequence $\{v_k^{-1} h_k\}$ is bounded, we assume, taking subsequence if necessary, that $\lim_{k \in \bar{K}} \frac{c_k}{v_k} = \tilde{y}$. Then, by taking limit on the first row of (6), we have

$$
\nabla f(\tilde{x}) + \nabla c(\tilde{x}) \tilde{y} = 0.
$$

Hence, $\tilde{x}$ is a KKT point.

**Theorem 3.23.** Suppose that Algorithm 3 does not terminate finitely, and that Assumptions 3.1 hold.

1. If $|K_h| = \infty$, then any accumulation point of $\{x_k\}_{k \in K_h}$ is a KKT point for (1).
2. If $|K_h| < \infty$, then any accumulation point of $\{x_k\}$ is a KKT point for (1).

4. **Local convergence.** In this section, we modify the algorithm presented above in order to achieve fast local convergence near a solution under certain assumptions.

Algorithm 3 itself is globally convergent as we have shown. However, in order to meet the condition (9) or (10), the algorithm may have to choose a small regularization parameter $v_k$, which causes slow convergence. Thus, in order to accelerate the algorithm near a solution, we should make some modifications on it.

The major idea of the modifications is to embed Algorithm 3 in a fast locally convergent algorithm. We use the following algorithm framework (Algorithm 4), which is a simplification of [1, Algorithm 1] that ignores inequality constraints.

Assume $x^*$ is a stationary point of (1) that satisfies (2). Let $\mathcal{Y}$ be the set of Lagrange multipliers associated to $x^*$, i.e.,

$$
\mathcal{Y} = \{y^* \in \mathbb{R}^n \mid (x^*, y^*) \text{ satisfies the KKT conditions (2)}\}.
$$

We use the following assumptions.

**Assumption 4.1.**

(B1) The sequence $(x_k, y_k)$ generated by Algorithm 4 converges to $(x^*, y^*)$ for a certain $y^* \in \mathcal{Y}$. 

Algorithm 4 Modified Algorithm

Initialization. Choose $\epsilon > 0$, $\beta > 0$, $\beta_1 \in (0, 1]$ and $\theta \in (0, 1)$. Set $k := 0$. 

repeat

Let $v_{k}^{\text{out}} = \|\nabla_x L_k\| + \|c_k\|$ and $\epsilon_k = (v_{k}^{\text{out}})^{1+\beta}$. Choose a proper $\delta_k > 0$. 

Compute $(d_k, \Delta y_k)$ such that

$$
(H_k + \delta_k I \nabla c_k \nabla c_k^T - v_{k}^{\text{out}} I \Delta y_k) (d_k, \Delta y_k) = - (\nabla_x L_k) .
$$

(56)

if $\|\nabla_x L(x_k + d_k, y_k + \Delta y_k)\| + \|c(x_k + d_k)\| \leq \theta(\|\nabla_x L_k\| + \|c_k\|) + \epsilon_k$, then
let $x_{k+1} = x_k + d_k$, $y_{k+1} = y_k + \Delta y_k$. Set $k := k + 1$.

else
Use Algorithm 3 to find a new iterate $(x_{k+1}, y_{k+1})$ such that

$\|\nabla_x L_{k+1}\| + \|c_{k+1}\| \leq \theta(\|\nabla_x L_k\| + \|c_k\|) + \epsilon_k$.

Set $k := k + 1$.

end if

until $\|\nabla_x L_k\| + \|c_k\| \leq \epsilon$

(B2) The functions $f$ and $c$ are twice continuously differentiable with locally Lipschitz second derivatives on $\mathbb{R}^n$.

(B3) The symmetric matrix $H_k$ is uniformly bounded.

(B4) The approximation $H_k$ is sufficiently positive definite on the null space of $\nabla c(x^*)^T$ for all sufficiently large $k$, i.e., there exists $\eta > 0$, such that $p^T H_k p \geq \eta \|p\|^2$ for all $p \in \text{Null}(\nabla c(x^*)^T)$.

(B5) $H_k$ satisfies

$$
\|(H_k - \nabla^2_{xx} L(x^*, y^*)) d_k\| = o(\|d_k\|)
$$

for all sufficiently large $k$, where $d_k$ is computed from (56).

Thus, the analysis of local convergence follows that of [1, 2, 31] based on these assumptions. We have

Theorem 4.2. Suppose that Assumption 4.1 holds. Then the sequence $(x_k, y_k)$ generated by Algorithm 4 converges $Q$-superlinearly to $(x^*, y^*)$.

5. Numerical experience. In this section, we report the performance of Algorithm 4 on the degenerate problems of DEGEN collection [10], compared with the well-known implementations MINOS and LOQO [52, 53].

5.1. An adaptively regularized interior point method. Since most of the test problems in DEGEN have inequality constraints, we firstly modify the new algorithm so that it can be applied to nonlinear optimization problems of the standard form

$$
\begin{align*}
\min & \ f(x) \\
\text{s.t.} & \ c(x) = 0, \\
& \ x \geq 0.
\end{align*}
$$

(57)

Interior point method is an efficient way for handling inequality constraints. Now we demonstrate how the new algorithm presented in Section 2 can be used within
an interior point framework. In general, an interior point method approximately solves a sequence of barrier problems

\[
\min \varphi^\mu(x) \triangleq f(x) - \mu \sum_{i=1}^n \ln x^{(i)}
\]

s.t. \( c(x) = 0 \),

with a positive barrier parameter \( \mu \), which is decreasing and converges to 0. An approximate solution \((x, y, z)\) is required to satisfy

\[
E^\mu(x, y, z) \leq \kappa_\epsilon \mu,
\]

where \( E^\mu(x, y, z) = \min\{\|\nabla f + \nabla cy - z\|_\infty, h(x), \|Xz - \mu e\|_\infty\} \) for some fixed \( \kappa_\epsilon > 0 \). The main scheme of the adaptively regularized interior point method for (57) is given in Algorithm 5:

**Algorithm 5** Adaptively regularized interior point method

Choose an initial value for the barrier parameter \( \mu > 0, \kappa_\epsilon, \kappa_\mu, \kappa_\tau \in (0, 1) \) and final step tolerance \( \epsilon \). Choose the starting point \((x, z) > 0\) and \( y \in \mathbb{R}^m \).

repeat

Apply Algorithm 3 or 4 to find an approximate solution \((x^\mu, y^\mu, z^\mu)\) of the barrier problem (60) satisfying (59).

Set \( \mu \leftarrow \max\{ \kappa_\mu E^\mu, \mu^2 \} \), \((x, y, z) \leftarrow (x^\mu, y^\mu, z^\mu)\).

until \( E_0(x, y, z) \leq \epsilon \)

The main computation lies in solving approximately the barrier problem (58). For a given point \( x_k \), a general interior point method gets a search direction from the solution of the following system

\[
\begin{pmatrix}
H_k + \Sigma_k + \delta_k I \\
\nabla c_k^T \\
\n\nabla c_k^T
\end{pmatrix}
\begin{pmatrix}
d \\
y_k \pm \\
\nabla c_k^T
\end{pmatrix}
= -
\begin{pmatrix}
\nabla f_k - \mu X^{-1}^k e \\
\nabla c_k^T \\
\nabla c_k^T
\end{pmatrix},
\]

where the matrix \( \Sigma_k \) takes \( \mu X^{-2} \) in primal methods, and equals \( X^{-1}_k Z_k \) where \( Z_k \) is a diagonal matrix with the elements of the dual variable \( z_k \) in primal dual methods. Our method adaptively regularizes the (60) as

\[
\begin{pmatrix}
H_k + \Sigma_k + \delta_k I \\
\nabla c_k^T \\
\n\nabla c_k^T
\end{pmatrix}
\begin{pmatrix}
d \\
y_k^+ \\
\nabla c_k^T
\end{pmatrix}
= -
\begin{pmatrix}
\nabla f_k - \mu X^{-1}_k e \\
\nabla c_k^T \\
\nabla c_k^T
\end{pmatrix}.
\]

To keep the components of the iterates bounded away from 0, step size \( \alpha \) should satisfy the following fraction-to-the-boundary rule

\[
x_k + \alpha d \geq (1 - \tau)x_k,
\]

where \( \tau \) is given by

\[
\tau = 1 - \min\{0.01, \kappa_\tau \mu\},
\]

where \( \kappa_\tau \) is a positive constant less than 1.

Wächter and Biegler [54] showed that, under certain assumptions, the components of \( x_k \) are bounded away from 0 for a fixed \( \mu \), as long as there exists \( \kappa_\Sigma > 1 \) such that the diagonal element \( \sigma_k^{(i)} \) of \( \Sigma_k \) satisfies

\[
\frac{1}{\kappa_\Sigma (x_k^{(i)})^2} \mu \leq \sigma_k^{(i)} \leq \frac{\kappa_\Sigma}{(x_k^{(i)})^2} \mu.
\]
for $i = 1, 2, \cdots, n$ and a proper restoration algorithm is used. Note that (62) is trivially true in primal interior point methods and is a commonly used safeguard technique in primal dual interior point methods, see [7, 58, 55] and etc. Hence, under the same assumptions as [54], the algorithm is still globally convergent when it is applied to interior point framework. Whether the similar convergence results under weaker assumptions still hold for adaptively regularized interior point methods is a topic for further work.

5.2. Calculating inertia. In Algorithm 1, the parameters $\delta_k$ and $v_k$ depend on the inertia of $K_k(\delta_k, v_k)$. Hence, calculating the inertia of a symmetric matrix is important for the implementation of the algorithm.

Suppose that $K$ is a matrix which is symmetric, but need not to be positive definite or quasi-definite [51]. Hence, $K$ may not be strongly factorable, i.e., it does not necessary have a factorization $PKPT = LDL^T$, where $P$ is a permutation matrix, $L$ is a unit lower-triangular matrix and $D$ is a diagonal matrix. Bunch and Kaufman [5] introduced some stable methods for calculating inertia of indefinite matrix. Following their ideas, we firstly do a block $LDL^T$ factorization for $K$:

$$PKPT = LDL^T$$

where $P$, $L$ are as above and $B$ is a block diagonal matrix, where each block is of order 1 or 2. Then $\text{In}(K) = \text{In}(B)$. Secondly, we scan the diagonal and lower sub-diagonal elements of $B$ to find the numbers of positive, negative and zero $1 \times 1$ blocks, and the number of $2 \times 2$ blocks. Suppose that $n_p$, $n_n$, $n_0$ and $n_2$ are numbers of them respectively. Bunch and Kaufman [5] pointed out that each $2 \times 2$ block corresponds to a positive-negative pair of eigenvalues. Therefore, we have

$$\text{In}(K) = \text{In}(B) = (n_p + n_2, n_n + n_2, n_0).$$

Block $LDL^T$ factorization of matrix $K_k(\delta_k, v_k)$ can be obtained from the course of solving (6) via a direct solver for symmetric indefinite system such as MA57 [12]. Hence, the only extra cost we need is to scan diagonal and lower sub-diagonal.

Note that if $H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T$ is positive definite, then so is $H_k + \frac{1}{v_k} \nabla c_k \nabla c_k^T$ for all $v \in (0, v_k]$, which implies by Lemma 3.3, $\text{In}(K_k(\delta_k, v)) = (n, m, 0)$ that for all $v \in (0, v_k]$. Hence, we do not need to calculate $\text{In}(K_k(\delta_k, v))$ for $0 < v \leq v_k$ if $\text{In}(K_k(\delta_k, v_k))$ has been found to be $(n, m, 0)$.

5.3. Second order correction. Newton-like methods for constrained optimization may suffer from Maratos effect, in which full steps may be rejected even if arbitrarily near the solution. Some techniques to deal with this situation are suggested by researchers, see, for instance [6, 15, 21, 36, 40, 41, 50, 48, 28, 55] and etc., among which is the second order correction (SOC) technique [15, 36, 21, 55]. The SOC technique computes an additional step which further improves feasibility, and hence has better chance to be accepted. In our implementation, the SOC step is computed only if $h_k < 100$ and the first trial step $\alpha_k^{\max} d_k$ is rejected. An SOC step $d_k^{\text{soc}}$ is obtained from the following linear system [55]

$$\begin{pmatrix} H_k + \sum_k + \delta_k I & \nabla c_k \\ \nabla c_k^T & -v_k I \end{pmatrix} \begin{pmatrix} d_k^{\text{soc}} \\ y^{\text{soc}} \end{pmatrix} = - \begin{pmatrix} 0 \\ c(x_k + \alpha_k^{\max} d_k) \end{pmatrix}$$

where $\alpha_k^{\max}$ is the maximal step size satisfying (61). Note that the coefficient matrix of the SOC system is the same as that of (60). Thus additional matrix factorizations are avoid. If the hybrid step $\alpha_k^{\max} d_k + d_k^{\text{soc}}$ is also rejected, then the algorithm continue the line search along direction $d_k$. Since the SOC step is tried
at most once per iteration, the global convergence theorems established before still hold.

5.4. Preliminary setting. At a starting point \( x_0 \), or an iterate \( x_j \) in a restoration phase, the vector of Lagrangian multipliers \( y \) is obtained as a least-square solution of \( \nabla \mathcal{L}(x, y) \), i.e., by solving the linear system
\[
\begin{pmatrix}
I & \nabla c \\
\nabla c^T & 0
\end{pmatrix}
\begin{pmatrix}
\xi \\
y
\end{pmatrix}
= -
\begin{pmatrix}
\nabla \varphi_0 \\
0
\end{pmatrix},
\]
(63)
and discarding \( \xi \). In our implementation, \( H_k = \nabla^2_{xx} \mathcal{L}_k \).

We use the following parameter setting in our implementation
\[
\begin{align*}
\nu &= 10^{-5}, \ \hat{v} = 10^{-3}, \ \gamma_v = 0.1, \ \gamma_1 = 0.5, \ \gamma_2 = 0.05, \ \nu_{\min} = 10^{-10}, \\
\gamma_{\nu_{\min}} &= 0.1, \ \sigma_v = 2.5, \ \rho = 10^{-5}, \ \alpha = 10^{-5}, \ \gamma_\alpha = 0.1, \\
\sigma_\alpha &= 2, \ \gamma_p = 10^{-10}, \ \kappa_h = 0.5, \ \kappa_{\bar{h}} = 0.25, \ \sigma_r = 2, \ \kappa_r = 0.5, \\
\kappa_e &= 10, \ \kappa_\mu = 0.25, \ \kappa_\tau = 0.01.
\end{align*}
\]

5.5. Numerical results. Our implementation is written in Matlab, combined with the use of MA57. The latter provides an effective solver for sparse symmetric linear systems. Regarding MINOS and LOQO, we use the user interfaces via AMPL modelling language [19], with the default values of all parameters except for the following. We increase the maximum number of major iteration of MINOS from 100 to 1500, and increase the limit of iterations of LOQO from 200 to 1500.

DEGEN is an AMPL collection which now contains 111 test problems with degenerate constraints. Most problems are very small, but many are in some ways difficult. We use all of them except for the following 5 instances: problems 20205 is unbounded, problems 2DD01-500h, 2DD01-500v, 2DD01-1000h and 2DD01-1000v are too large compared to the other problems.

In DEGEN AMPL models, starting point is randomly chosen in a domain of a specified size. For the convenience of comparison, we use a fixed starting point \((10, \ldots, 10)\) for most of the problems except for problems 20212 and 20215, because \((10, 10)\) is a solution of them. For these two problems, the algorithms start at \((10, 5)\). The new algorithm computes initial Lagrangian multipliers by (63) while MINOS and LOQO use there default multiplier settings.

Note that MINOS and LOQO use different stopping rules, see [39, 52, 53]. Though these rules are equivalent in theory, they do have some different impacts on the performance of an algorithm. In order to put the solvers into more-or-less similar conditions, we compare the new algorithm to these two implementations respectively, using as similar stopping rules to that of the compared method as possible in each comparison. When compared to LOQO, the algorithm uses the termination criteria that LOQO used, which are that the primal and dual agree to 8 significant figures and that the primal and dual are feasible to the \(10^{-6}\) relative error level. When compared to MINOS, the algorithm is terminated if both the measure of optimality and the measure of feasibility are less than \(10^{-6}\).

The results are summarized in Tables 1 and 2. In these two tables, we summarize the number of problems solved within 1500 function evaluations, the robustness of the solvers, the average and median function evaluations the solvers used for those problems solved within 1500 function evaluations.

Next, we report the comparison of performance of these algorithms in the form of performance profiles proposed by Dolan and Moré [11]. Suppose that we are comparing the performance of \(n_s\) solvers on \(n_p\) problems. Let \(t_{p,s}\) denote the function
evaluations to solve problem $p$ by solver $s$. Then the performance ratio of solver $s$ on problem $p$ is defined as

$$ r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : 1 \leq s \leq n_s\}}. $$

If a solver $\tilde{s}$ does not solve problem $\tilde{p}$, then the ratio $r_{\tilde{p},\tilde{s}}$ is assigned

$$ r_{\tilde{p},\tilde{s}} = 2 \max\{r_{p,s} : \text{solver } s \text{ solves problem } p\}. $$

Then the logarithmic performance function for each solver is defined as

$$ \rho_s(\tau) = \frac{\text{number of problems s.t. } \log_2(r_{p,s}) \leq \tau}{n_p}. $$

We compare in Figures 1 and 2 the relative performance of the new algorithm compared to MINOS and LOQO in terms of function evaluations. Observe that the new algorithm performs better than MINOS and LOQO on DEGEN collection.

6. **Final remarks.** We present an adaptively regularization SQP scheme for equality constrained optimization. This method is shown to be globally convergent and is locally convergent after certain modifications. Preliminary numerical results are encouraging. The main ideas of this algorithm can be generalized to general constrained optimization problems. Further work mainly focuses on two aspects: (1) generalizing the framework to other problems, such as mathematical programs with equilibrium constraints, nonlinear complementarity problems and etc. (2) practical
considerations so as to developing a robust and efficient implementation for large scale problems.

Acknowledgement. We would like to thank the Editor and the anonymous referees for their valuable and helpful comments that have improved the quality of this paper greatly.

REFERENCES

[1] P. Armand, J. Benoist and D. Orban, From global to local convergence of interior methods for nonlinear optimization, Optim. Method. Softw., 28 (2013), 1051–1080.

[2] S. Arreckx and D. Orban, A Regularized Factorization-Free Method for Equality-Constrained Optimization, Technical Report G-2016-65, GERAD HEC Montréal, 2016.

[3] R. H. Biefskowsky and F. A. M. Gomes, Dynamic control of infeasibility in equality constrained optimization, SIAM J. Optim., 19 (2008), 1299–1325.

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

[5] J. R. Bunch and L. Kaufman, Some stable methods for calculating inertia and solving symmetric linear systems, Math. Comput., 31 (1977), 163–179.

[6] R. M. Chamberlain, M. J. D. Powell, C. Lemarechal and H. C. Pedersen, The watchdog technique for forcing convergence in algorithms for constrained optimization, in Algorithms for Constrained Minimization of Smooth Nonlinear Functions (eds. A. G. Buckley and J. L. Goffin), Springer, Berlin, Heidelberg, 1982, 1–17.

[7] A. R. Conn, N. I. M. Gould, D. Orban and P. L. Toint, A primal-dual trust-region algorithm for non-convex nonlinear programming, Math. Program., 87 (2000), 215–249.

[8] A. Conn, N. Gould and P. Toint, Trust-Region Methods, SIAM, Philadelphia, PA, USA, 2000.

[9] H. Dan, N. Yamashita and M. Fukushima, Convergence properties of the inexact Levenberg-Marquardt method under local error bound conditions, Optim. Method. Softw., 17 (2002), 605–626.

[10] DEGEN, Http://w3.impa.br/~optim/DEGEN_collection.zip.

[11] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, Math. Program., 91 (2002), 201–213.

[12] I. S. Duff, Ma57—a code for the solution of sparse symmetric definite and indefinite systems, ACM Trans. Math. Softw., 30 (2004), 118–144.

[13] J.-Y. Fan and Y.-X. Yuan, On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption, Computing, 74 (2005), 23–39.

[14] D. Fernández, E. A. Pilotta and G. A. Torres, An inexact restoration strategy for the globalization of the sQP method, Comput. Optim. Appl., 54 (2013), 595–617.

[15] R. Fletcher, Second order corrections for non-differentiable optimization, in Numerical Analysis: Proceedings of the 9th Biennial Conference Held at Dundee, Scotland, June 23–26, 1981 (ed. G. A. Watson), Springer, Berlin, Heidelberg, 1982, 85–114.

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

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

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

[19] R. Fourer, D. M. Gay and B. W. Kernighan, AMPL: A Modeling Language for Mathematical Programming, The Scientific Press, 1993.

[20] M. P. Friedlander and D. Orban, A primal–dual regularized interior-point method for convex quadratic programs, Math. Program. Comput., 4 (2012), 71–107.

[21] M. Fukushima, A successive quadratic programming algorithm with global and superlinear convergence properties, Math. Program., 35 (1986), 253–264.

[22] P. E. Gill, W. Murray, D. Ponceleón and M. Saunders, Solving Reduced KKT Systems in Barrier Methods for Linear and Quadratic Programming, Technical Report Technical Report SOL 91-7, Systems Optimization Laboratory, Stanford University, Stanford, 1991.

[23] P. E. Gill, V. Kungurtsev and D. P. Robinson, A stabilized SQP method: Superlinear convergence, Math. Program., 163 (2017), 369-410.
2700 SONGQIANG QIU AND ZHONGWEN CHEN

[24] P. E. Gill and D. P. Robinson, A globally convergent stabilized SQP method, *SIAM J. Optim.*, **23** (2013), 1983–2010.

[25] P. E. Gill and E. Wong, Sequential quadratic programming methods, in *Mixed Integer Nonlinear Programming* (eds. J. Lee and S. Leyffer), Springer, New York, 2012, 147–224.

[26] J. Gondzio, Matrix-free interior point method, *Comput. Optim. Appl.*, **51** (2012), 457–480.

[27] N. I. M. Gould and Ph. L. Toint, Nonlinear programming without a penalty function or a filter, *Math. Program.*, **122** (2010), 155–196.

[28] N. I. M. Gould and Ph. L. Toint, Global convergence of a non-monotone trust-region filter algorithm for nonlinear programming, in *Multiscale Optimization Methods and Applications* (eds. W. W. Hager, S.-J. Huang, P. M. Pardalos and O. A. Prokopyev), Springer US, Boston, MA, 2006, 125–150.

[29] W. W. Hager, Stabilized sequential quadratic programming, *Comput. Optim. Appl.*, **12** (1999), 253–273.

[30] A. F. Izmailov, M. V. Solodov and E. I. Uskov, Combining stabilized SQP with the augmented Lagrangian algorithm, *Comput. Optim. Appl.*, **62** (2015), 405–429.

[31] A. F. Izmailov and M. V. Solodov, Stabilized SQP revisited, *Math. Program.*, **133** (2012), 93–129.

[32] K. Levenberg, A method for the solution of certain problems in least squares, *Q. Appl. Math.*, **2** (1944), 164–168.

[33] D.-H. Li and L. Qi, A Stabilized SQP Method via Linear Equations, Applied mathematics technical report AMR00/5, The University of New South Wales, 2000.

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

[35] C. M. Maes, *A Regularized Active-Set Method for Sparse Convex Quadratic Programming*, PhD thesis, Institute for Computational and Mathematical Engineering, Stanford University, CA, 2010.

[36] D. Q. Mayne and E. Polak, A superlinearly convergent algorithm for constrained optimization problems, in *Algorithms for Constrained Minimization of Smooth Nonlinear Functions* (eds. A. G. Buckley and J.-L. Goffin), Springer Berlin Heidelberg, Berlin, Heidelberg, 1982, 45–61.

[37] L. Minchenko and S. Stakhovski, On relaxed constant rank regularity condition in mathematical programming, *Optimization*, **60** (2011), 429–440.

[38] J. J. Moré, The Levenberg-Marquardt algorithm: Implementation and theory, in *Numerical Analysis*, Springer, Berlin, 1978, 105–116.

[39] B. A. Murtagh and M. A. Saunders, *Minos 5.4 User’s Guide (Revised)*, Technical report, Technical Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA 94305, USA, 1993; Revised, 1995.

[40] M. J. D. Powell and Y.-X. Yuan, A recursive quadratic programming algorithm that uses differentiable exact penalty functions, *Math. Program.*, **35** (1986), 265–278.

[41] M. J. D. Powell and Y.-X. Yuan, A trust region algorithm for equality constrained optimization, *Math. Program.*, **49** (1990), 189–211.

[42] S. Q. Qiu and Z. W. Chen, Global and local convergence of a class of penalty-free-type methods for nonlinear programming, *Appl. Math. Model.*, **36** (2012), 3201–3216.

[43] S. Q. Qiu and Z. W. Chen, A globally convergent penalty-free method for optimization with equality constraints and simple bounds, *Acta Appl. Math.*, **142** (2016), 39–66.

[44] T. Rusten and R. Winther, A preconditioned iterative method for saddlepoint problems, *SIAM J. Matrix Anal. Appl.*, **13** (1992), 887–904.

[45] M. Saunders and J. Tomlin, *Solving Regularized Linear Programs Using Barrier Methods and KKT Systems*, Technical Report SOL Report 96-4, Dept. of EESOR, Stanford University, 1996.

[46] D. Silvester and A. Wathen, Fast iterative solution of stabilised Stokes systems part II: Using general block preconditioners, *SIAM J. Numer. Anal.*, **31** (1994), 1352–1367.

[47] W. Y. Sun and Y.-X. Yuan, *Optimization Theory and Methods*, Springer, 2005.

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

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

[50] S. Ulbrich, On the superlinear local convergence of a filter-SQP method, *Math. Program.*, **100** (2004), 217–245.
[51] R. J. Vanderbei, Symmetric quasidefinite matrices, *SIAM J. Optim.*, 5 (1995), 100–113.
[52] R. J. Vanderbei, Loqo: An interior point code for quadratic programming, *Optim. Methods Softw.*, 11 (1999), 451–484.
[53] R. J. Vanderbei and D. F. Shanno, An interior-point algorithm for nonconvex nonlinear programming, *Comput. Optim. Appl.*, 13 (1999), 231–252.
[54] A. Wächter and L. T. Biegler, Line search filter methods for nonlinear programming: Motivation and global convergence, *SIAM J. Optim.*, 16 (2005), 1–31 (electronic).
[55] A. Wächter and L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Program.*, 106 (2006), 25–57.
[56] S. J. Wright, Superlinear convergence of a stabilized SQP method to a degenerate solution, *Comput. Optim. Appl.*, 11 (1998), 253–275.
[57] W. J. Xue, C. G. Shen and D. G. Pu, A penalty-function-free line search SQP method for nonlinear programming, *J. Comput. Appl. Math.*, 228 (2009), 313–325.
[58] H. Yamashita, H. Yabe and T. Tanabe, A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization, *Math. Program.*, 102 (2005), 111–151.
[59] N. Yamashita and M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, in *Topics in Numerical Analysis*, Springer, 2001, 239–249.
[60] J.-L. Zhang, On the convergence properties of the Levenberg-Marquardt method, *Optimization*, 52 (2003), 739–756.

Received June 2018; revised March 2019.

E-mail address: sqiu@cumt.edu.cn
E-mail address: zwchen@suda.edu.cn